

**IMPLEMENTASI MODEL KLASIFIKASI SPESIES *STINGLESS BEE*  
MENGGUNAKAN MODEL CNN MOBILENETV2 DAN  
EFFICIENTNETB0 (STUDI KASUS: DATA CITRA LEBAH MADU  
DI LEMBAH SUHITA)**

**Skripsi**

**Oleh  
ANNISA ZHAFIRAH  
2115061038**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2025**

**IMPLEMENTASI MODEL KLASIFIKASI SPESIES *STINGLESS BEE*  
MENGGUNAKAN MODEL CNN MOBILENETV2 DAN  
EFFICIENTNETB0 (STUDI KASUS: DATA CITRA LEBAH MADU  
DI LEMBAH SUHITA)**

Oleh

**ANNISA ZHAFIRAH**

**Skripsi**

**Diajukan untuk Memenuhi Kelulusan Gelar SARJANA TEKNIK  
pada  
Program Studi Teknik Informatika  
Jurusan Teknik Elektro  
Fakultas Teknik**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2025**

## **ABSTRAK**

### **IMPLEMENTASI MODEL KLASIFIKASI SPESIES *STINGLESS BEE* MENGGUNAKAN MODEL CNN MOBILENETV2 DAN EFFICIENTNETB0 (STUDI KASUS: DATA CITRA LEBAH MADU DI LEMBAH SUHITA)**

**Oleh**  
**Annisa Zhafirah**

Lebah madu tanpa sengat (*stingless*) memiliki potensi yang unggul dibanding lebah sengat pada umumnya dalam perannya sebagai penghasil produk-produk lebah seperti madu, selain itu peranan lebah secara umum sangat penting dalam keberlanjutan ekosistem. Namun, upaya dalam pelestariannya masih tergolong minim, disisi lain tantangan lain yang dihadapi salah satunya masih kurangnya pengetahuan masyarakat dalam identifikasi spesies lebah *stingless* karena visual morfologi lebah yang beragam dan hampir serupa. Maka dari itu, penelitian ini bertujuan membuat sebuah model klasifikasi citra dari empat jenis (*spesies*) lebah *stingless* menggunakan metode *deep learning* yaitu *Convolutional Neural Network* (CNN). Data yang digunakan berjumlah 800 data yang diambil langsung di Lembah Suhita. Arsitektur CNN diterapkan dengan proses *transfer learning* pada *pretrained* model MobileNetV2 dan EfficientNetB0. Hasil penelitian menunjukkan akurasi terbaik pada model MobileNetV2 mencapai dua kali lipat akurasi dibanding model EfficientNetB0, dengan akurasi tertinggi sebesar 91% setelah proses *fine tuning*.

**Kata kunci:** Lebah Tanpa Sengat, *Convolutional Neural Network* (CNN), Klasifikasi, *Transfer Learning*, *Fine Tuning*.

## **ABSTRACT**

***IMPLEMENTATION OF STINGLESS BEE SPECIES CLASSIFICATION  
MODEL USING CNN MOBILENETV2 AND EFFICIENTNETB0  
(CASE STUDY: HONEY BEE IMAGES DATA IN SUHITA VALLEY)***

by  
**Annisa Zhafirah**

Stingless bees have superior potential compared to common stinging bees in their role as producers of bee products, such as honey. Furthermore, the general role of bees is crucial for ecosystem sustainability. However, conservation efforts for these bees remain limited. Another challenge is the lack of public knowledge in identifying stingless bee species due to their diverse and visually similar morphology. Therefore, this research aims to build an image classification model for four species of stingless bees using a Convolutional Neural Network (CNN). The dataset used consists of 800 images collected directly at Lembah Suhita. The CNN architecture was implemented using a transfer learning approach on the pre-trained MobileNetV2 and EfficientNetB0 models. The results showed that the best-performing model, MobileNetV2, achieved twice the accuracy of the EfficientNetB0 model, reaching a peak accuracy of 91% after a fine-tuning.

***Keywords:*** *Stingless Bee, Convolutional Neural Network (CNN), Classification, Transfer Learning, Fine Tuning.*

Judul Skripsi : **IMPLEMENTASI MODEL KLASIFIKASI SPESIES STINGLESS BEE MENGGUNAKAN MODEL CNN MOBILENETV2 DAN EFFICIENTNETB0 (STUDI KASUS: DATA CITRA LEBAH MADU DI LEMBAH SUHITA)**

Nama Mahasiswa : **Annisa Zahfirah**

Nomor Pokok Mahasiswa : **2115061038**

Program Studi : **Teknik Informatika**

Jurusan : **Teknik Elektro**

Fakultas : **Teknik**

Pembimbing Utama

**Yessi Mulyani, S.T., M.T.**  
**NIP. 197312262000122001**

Pembimbing Pendamping

**Rio Ariesia Pradipta, S.Kom., M.T.I.**  
**NIP. 198603232019031013**

2. Mengetahui

Ketua Jurusan Teknik Elektro

**Herlinawati, S.T., M.T.**  
**NIP. 197103141999032001**

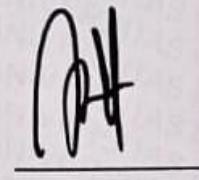
Ketua Program Studi Teknik Informatika

**Yessi Mulyani, S.T., M.T.**  
**NIP. 197312262000122001**

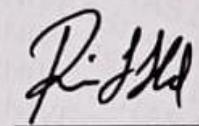
**MENGESAHKAN**

1. Tim Penguji

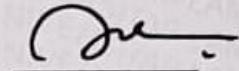
Ketua : **Yessi Mulyani, S.T., M.T.**



Sekretaris : **Rio Arestia Pradipta, S.Kom., M.T.I.**



Penguji : **Dr.Ir.M.Komaruddin, S.T., M.T.**



2. Dekan Fakultas Teknik



 **Dr.Eng. Ir. Helmy Fitriawan, S.T., M.Sc.**

**NIP. 197509282001121002**

Tanggal Lulus Ujian Skripsi: **25 Juli 2025**

## SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "**Implementasi Model Klasifikasi Spesies *Stingless Bee* Menggunakan Model CNN MobileNetV2 dan EfficientNetB0 (Studi Kasus: Data Citra Lebah Madu di Lembah Suhita)**" dibuat oleh saya sendiri. Semua hasil yang tertuang pada skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung.

Atas pernyataan ini, apabila dikemudian hari ternyata ditemukan adanya ketidakbenaran dari pernyataan saya, maka saya bersedia menanggung akibat dan dikenai sanksi sesuai hukum yang berlaku.

Bandar Lampung, 30 Juli 2025  
Pembuat Pernyataan



## RIWAYAT HIDUP



Penulis dilahirkan di Bengkulu, pada tanggal 07 September 2003 sebagai anak pertama dari dua bersaudara, dengan Orang Tua bernama Bapak Mirwansyah, S.E. dan Ibu Henny Silatri, S.E. Pendidikan Taman Kanak-Kanak (TK) ditempuh di TK. Quratta A'yun yang diselesaikan pada tahun 2009. Sekolah Dasar (SD) ditempuh di SD Negeri 05 Kabupaten Bengkulu Selatan pada tahun 2009-2015. Sekolah Menengah Pertama (SMP) ditempuh di SMP Negeri 01 Kabupaten Bengkulu Selatan dari tahun 2015-2018. Sekolah Menengah Atas (SMA) ditempuh di SMA Negeri 01 Kabupaten Bengkulu Selatan dari tahun 2018-2021. Selama masa sekolah penulis pernah mengikuti beberapa ajang seperti Cerdas Cermat Tkt. Provinsi, Vocal Grup Tkt. Provinsi, Olimpiade Sains Nasional (OSN) Matematika Tkt. Provinsi, dan Peraih tiga besar nilai Ujian Nasional (UN) SMP Tkt. Kabupaten.

Tepat pada bulan Agustus tahun 2021 penulis terdaftar sebagai salah satu mahasiswa Program Studi Teknik Informatika Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN). Selama masa perkuliahan penulis mengikuti kegiatan kemahasiswaan jurusan yaitu Himpunan Mahasiswa Teknik Elektro (HIMATRO) pada periode 2022 dan 2023. Penulis mengambil konsentrasi bidang *Artificial Intelligence* dari empat pilihan konsentrasi yang tersedia. Pada tahun 2023 penulis mengikuti program Kampus Merdeka yaitu *Studi Independent* pada mitra REVO-U dengan bidang *Software Engineering* dan *Data Analist* yang berlangsung dari bulan Juli hingga Desember. Pada Januari 2024 penulis menjalani Kuliah Kerja Nyata (KKN) periode I di Desa Way Tuba Kabupaten Way Kanan, Lampung. Lalu pada 28 Juni hingga 3 Agustus tahun 2024 penulis juga menjalani Kerja Praktik (KP) di PT. Telkom witel Lampung selama 40 hari.

## MOTTO

“Setiap kali kamu mencoba, kamu telah mengambil 7/10 peluang untuk berhasil. Sekalipun 7/10 dapat gagal, jangan biarkan ketakutan terlebih dahulu membatasi potensimu.”

*(Zhaf)*

“Sesulit apapun pulanglah sebagai seorang Sarjana dan sampaikan kepada dua orang terpenting yang selalu menunggu kepulanganmu bahwa mereka hebat.”

*(Zhaf)*

“Barang siapa yang mengerjakan kebaikan sekecil apapun, niscaya dia akan melihat (balasan)nya.”

*(Q.S Al-Zalzalah: 7)*

“Everything will be good, as long as you do your best. Because if you do, there will be no regrets.”

*(Tiffany Young)*

## **PERSEMPAHAN**

Alhamdulillahirabbil'alamin Segala puji bagi Allah SWT. Tuhan Yang Maha Esa, karena atas segala nikmat, rahmat, dan karunia-Nya, penulis dapat menyelesaikan skripsi ini tepat pada waktunya. Penulis persembahkan skripsi ini kepada:

Kedua Orang Tua, Adik, dan seluruh Keluarga yang telah memberikan doa, harapan, dan semangat yang tiada hentinya sehingga penulis dapat menuntut ilmu hingga saat ini. Semoga Allah SWT. senantiasa memberikan kebahagian dan keselamatan keluarga kita dunia dan akhirat.

Semua dosen program studi Teknik Informatika yang telah memberikan arahan dan ajaran baik dalam banyak hal. Semoga Allah SWT. senantiasa membalas kebaikan atas ilmu yang kalian berikan dengan kesabaran dan pengorbanan.

Sahabat, partner, dan rekan seperjuangan yang telah memberikan support, motivasi, dan bantuan selama masa perkuliahan. Semoga Allah SWT. Selalu melindungi kalian semua dan diberikan kelancaran untuk hal kedepannya.

Untuk Almamater tercinta Universitas Lampung.  
Terima kasih untuk pengalaman dan waktunya selama kurang lebih 4 tahun telah mendukung penulis dalam menempuh pendidikan sarjana ini.

*Jazakumullah Khairan Katsiran Wa Jazakumullah Ahsanal Jaza.*

## SANWACANA

Alhamdulillah segala puji dan syukur bagi Allah SWT. atas berkat rahmat dan karunia-Nya sehingga Laporan Kerja Praktik dengan judul "**Implementasi Model Klasifikasi Spesies *Stingless Bee* Menggunakan Model CNN MobileNetV2 dan EfficientNetB0 (Studi Kasus: Data Citra Lebah Madu di Lembah Suhita)**" dapat diselesaikan meski masih terdapat beberapa kekurangan.

Terselesaikannya laporan kerja praktik ini tentu tidak terlepas dari dukungan, bantuan, bimbingan, dan pengarahan dari berbagai pihak. Untuk itu, ungkapan rasa terima kasih setinggi-tingginya disampaikan kepada:

1. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung.
2. Ibu Herlinawati, S.T.,M.T. selaku Ketua Jurusan Teknik Elektro, Universitas Lampung.
3. Ibu Yessi Mulyani S.T.,M.T., selaku Ketua Program Studi Teknik Informatika, Universitas Lampung sekaligus sebagai Dosen Pembimbing Utama atas kesediaan waktu, arahan, dorongan, motivasi, dan bimbingannya.
4. Bapak Rio Ariestia Pradipta, S.Kom, M.T.I. selaku Dosen Pembimbing Pendamping, atas kesediaan waktunya memberikan bimbingan dan pengarahan dalam penelitian ini.
5. Bapak Ir.M. Komarudin, S.T.,M.T. selaku Dosen Penguji yang telah bersedia memberi banyak masukan dan arahan perbaikan kepada penulis selama melaksanakan penelitian.
6. Ibu Ir. Trisy Septiana, S.T.,M.T., IPM. selaku Dosen Pembimbing Akademik yang telah bersedia memberi motivasi dan masukan kepada penulis selama melaksanakan penelitian dan proses perkuliahan.
7. Staf Administrasi Teknik Informatika Universitas Lampung yang telah membantu dapat urusan administrasi dan surat menyurat.

8. Pimpinan Lembah Suhita yang telah memberikan izin kepada peneliti untuk melakukan penelitian dan mengambil data penelitian di lokasi Lembah Suhita.
9. Papa, Mama, dan Adik selaku keluarga yang tak pernah absen memberikan dorongan semangat, nasihat, dan do'a untuk menyelesaikan penelitian ini.
10. Fadila Oktavianti dan Arifal Gusman selaku sahabat sekaligus orang terdekat yang selalu memberikan semangat, dukungan, dan dorongan agar penelitian ini dapat diselesaikan.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam penelitian ini, baik dari segi penulisan maupun pengembangan model yang dibuat. Oleh sebab itu, penulis mengharapkan saran dan kritik yang membangun terkait penelitian ini. Penulis juga berharap semoga penelitian ini bisa memberikan manfaat yang baik untuk pembaca maupun semua pihak terkait.

Bandar Lampung, 30 Juli 2025



**Annisa Zhafirah**

## DAFTAR ISI

	Halaman
<b>DAFTAR ISI.....</b>	<b>i</b>
<b>DAFTAR TABEL .....</b>	<b>iv</b>
<b>DAFTAR GAMBAR.....</b>	<b>vi</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>viii</b>
<b>I. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	4
1.5 Batasan Masalah .....	4
1.6 Sistematika Penulisan .....	5
<b>II. TINJAUAN PUSTAKA.....</b>	<b>7</b>
2.1 Dasar Teori.....	7
2.1.1 <i>Spesies Lebah Madu</i> .....	7
2.1.2 <i>Artificial Intelligence</i> .....	10
2.1.3 <i>Artificial Neural Network</i> .....	12
2.1.4 <i>Deep Learning</i> .....	15
2.1.5 Convolutional Neural Network (CNN).....	16
2.1.6 Backpropagation .....	22
2.1.7 <i>Supervised Learning</i> .....	25
2.1.8 Citra Digital.....	25
2.1.9 Dataset.....	27
2.1.10 <i>Library dan Framework</i> .....	27
2.1.11 Teknik Augmentasi .....	28
2.1.12 Validasi dan Optimalisasi .....	28
2.1.13 Metrik Evaluasi .....	29
2.2 Penelitian Terdahulu .....	31

2.3	Analisis Penelitian Serupa .....	36
2.4	Tools yang Digunakan .....	37
2.4.1	Jupyter Notebook .....	37
2.4.2	Google Colab .....	37
2.4.3	Python .....	38
2.4.4	TensorFlow dan Keras .....	39
2.4.5	Scikit-Learn.....	40
2.4.6	GitHub.....	40
2.4.7	Draw.io.....	41
2.4.8	Canva.....	41
2.4.9	Google Drive.....	42
<b>III.</b>	<b>METODE PENELITIAN .....</b>	<b>43</b>
3.1	Waktu dan Tempat Penelitian.....	43
3.1.1	Waktu Penelitian .....	43
3.1.2	Tempat Penelitian.....	43
3.2	Alat dan Bahan Penelitian.....	44
3.2.1	Alat Penelitian.....	44
3.2.2	Bahan Penelitian.....	45
3.3	Tahapan Penelitian.....	45
3.3.1	Planning (Perencanaan).....	46
3.3.2	Data <i>Collect</i> (Pengumpulan Data) .....	46
3.3.3	Eksplorasi dan <i>Pre-processing</i> Data .....	47
3.3.4	Augmentasi Data .....	48
3.3.5	Implementasi <i>Pretrained CNN</i> Model.....	48
3.3.6	Pelatihan ( <i>Training</i> ) Model.....	49
3.3.7	Evaluasi Model.....	50
<b>IV.</b>	<b>PEMBAHASAN .....</b>	<b>51</b>
4.1	Pengumpulan Dataset.....	51
4.2	Preprocessing Dataset .....	52
4.2.1	Foldering Data.....	52
4.2.2	Splitting dan Renaming Data .....	52
4.2.3	Resizing Data .....	54

4.3 Implementasi Model .....	55
4.3.1 Arsitektur MobileNetV2 .....	62
4.3.2 Arsitektur EfficientNetB0 .....	75
4.4 Penentuan Model Terbaik .....	83
4.5 Evaluasi Model Terbaik.....	84
4.6 Analisis Hasil Model Terbaik .....	98
4.7 Temuan Penelitian .....	100
<b>V. PENUTUP.....</b>	<b>102</b>
5.1 Kesimpulan .....	102
5.2 Saran .....	102
<b>DAFTAR PUSTAKA .....</b>	<b>103</b>
<b>LAMPIRAN.....</b>	<b>108</b>

## DAFTAR TABEL

	Halaman
Tabel 2. 1. Definisi AI dalam Sudut Pandang Karakteristik .....	10
Tabel 2. 2. Contoh Tabel <i>Confusion Matrix</i> .....	30
Tabel 3. 1. Waktu Pelaksanaan Penelitian .....	43
Tabel 3. 2. Perangkat Keras .....	44
Tabel 3. 3. Perangkat Lunak .....	44
Tabel 4. 1. Spesifikasi Perangkat .....	55
Tabel 4. 2. Spesifikasi Komputasi Latih Model.....	56
Tabel 4. 3. Variabel Teknik Augmentasi .....	58
Tabel 4. 4. Penentuan Batch Size.....	60
Tabel 4. 5. Penentuan Epoch Optimal.....	61
Tabel 4. 6. Rincian Model Summary Arsitektur MobileNetV2.....	64
Tabel 4. 7. Rangkuman MobileNetV2 .....	67
Tabel 4. 8. Rangkuman MobileNetV2 Fine Tuning .....	70
Tabel 4. 9. Penambahan Variabel Teknik Augmentasi.....	72
Tabel 4. 10. Rangkuman MobileNetV2 Fine Tuning + Var.Augmentasi.....	73
Tabel 4. 11. Rincian Model Summary EfficientNetB0.....	77
Tabel 4. 12. Rangkuman EfficientNetB0 .....	80
Tabel 4. 13. Rangkuman EfficientNetB0 Fine Tuning + Var.Augmentasi.....	82
Tabel 4. 14. Perbandingan <i>Training</i> MobileNetV2 dan EfficientNetB0 .....	83
Tabel 4. 15. <i>Classification Report</i> MobileNetV2 .....	85
Tabel 4. 16. <i>Classification Report</i> MobileNetV2 Fine Tuning.....	89
Tabel 4. 17. <i>Classification Report</i> MobileNetV2 Fine Tuning+Var.Augmentasi	92
Tabel 4. 18. <i>Classification Report</i> MobileNetV2 Fine Tuning Data Uji Baru .....	93
Tabel 4. 19. Hasil <i>Probability</i> Contoh Uji Citra Tunggal Apicalis .....	95
Tabel 4. 20. Hasil <i>Probability</i> Contoh Uji Citra Tunggal Biroi .....	96
Tabel 4. 21. Hasil <i>Probability</i> Contoh Uji Citra Tunggal Itama.....	97
Tabel 4. 22. Hasil <i>Probability</i> Contoh Uji Citra Tunggal Thoracica.....	98

Tabel 4. 23. Perbandingan Evaluasi Performa Model MobileNetV2 .....	99
Tabel 4. 24. Hasil Evaluasi Data Tunggal Lebah Lain .....	101
Tabel 4. 25. Hasil Evaluasi Data Tunggal Lainnya .....	101

## DAFTAR GAMBAR

	Halaman
Gambar 2. 1. Lebah Madu Stingless .....	8
Gambar 2. 2. Pohon Kelas Lebah yang Diteliti .....	9
Gambar 2. 3. Ilustrasi Neuron pada Jaringan Saraf Tiruan.....	12
Gambar 2. 4. Contoh Arsitektur Jaringan Saraf Tiruan .....	13
Gambar 2. 5. Arsitektur Layers pada Proses CNN .....	17
Gambar 2. 6. Arsitektur Base Model MobileNetV2 .....	20
Gambar 2. 7. Arsitektur Base Model EfficientNetB0 .....	21
Gambar 2. 8. Feedforward-Pass Algoritma Backpropagation .....	22
Gambar 2. 9. Backward-Pass Algoritma Backpropagation .....	24
Gambar 3. 1. Alur Kerja (Proses) Penelitian .....	45
Gambar 4. 1. Foldering Kelas Data.....	52
Gambar 4. 2. Contoh Kode Rename Setiap Kelas Data Latih .....	53
Gambar 4. 3. Hasil Renaming Setiap Kelas Data .....	53
Gambar 4. 4. Kode Resize Data Citra 224 x 224 .....	54
Gambar 4. 5. Perbandingan Citra Sebelum dan Sesudah Resizing.....	55
Gambar 4. 6. Kode Pustaka (library) Pembuatan Model .....	56
Gambar 4. 7. Kode Import dan Labeling Data.....	57
Gambar 4. 8. Arsitektur Model MobileNetV2 .....	62
Gambar 4. 9. Kode Program Pembuatan Model MobileNetV2.....	62
Gambar 4. 10. Kode Program Callbacks MobileNetV2 .....	65
Gambar 4. 11. Kode Program Training Model Fit.....	67
Gambar 4. 12. Grafik Accuracy MobileNetV2 .....	68
Gambar 4. 13. Grafik Loss MobileNetV2.....	69
Gambar 4. 14. Kode Program Fine Tuning MobileNetV2 .....	69
Gambar 4. 15. Grafik Accuracy MobileNetV2 Fine Tuning .....	71
Gambar 4. 16. Grafik Loss MobileNetV2 Fine Tuning .....	72
Gambar 4. 17. Grafik Accuracy MobileNetV2 Fine Tuning + Var.Augmentasi..	74

Gambar 4. 18. Grafik Loss MobileNetV2 Fine Tuning + Var.Augmentasi .....	74
Gambar 4. 19. Arsitektur Model EfficientNetB0.....	75
Gambar 4. 20. Kode Program Model EfficientNetB0 .....	75
Gambar 4. 21. Kode Program Callbacks EfficientNetB0 .....	78
Gambar 4. 22. Kode Program Training Model EfficientNetB0.....	79
Gambar 4. 23. Grafik Accuracy EfficientNetB0.....	80
Gambar 4. 24. Kode Program Fine Tuning Model EfficientNetB0.....	81
Gambar 4. 25. Grafik Loss EfficientNetB0 Fine Tuning + Var. Augmentasi .....	83
Gambar 4. 26. Confusion Matrix MobileNetV2 (Tanpa Fine Tuning).....	84
Gambar 4. 27. Confusion Matrix MobileNetV2 Fine Tuning .....	88
Gambar 4. 28. Confusion Matrix MobileNetV2 Fine Tuning + Var. Augmentasi	91
Gambar 4. 29. Confusion Matrix MobileNetV2 Fine Tuning Data Uji Baru .....	93
Gambar 4. 30. Grafik Distribusi Probabilitas Citra Tunggal Apicalis.....	95
Gambar 4. 31. Grafik Distribusi Probabilitas Citra Tunggal Biroi .....	96
Gambar 4. 32. Grafik Distribusi Probabilitas Citra Tunggal Itama .....	97
Gambar 4. 33. Grafik Distribusi Probabilitas Citra Tunggal Thoracica .....	98

## **DAFTAR LAMPIRAN**

- Lampiran 1. Sample Kelas Dataset
- Lampiran 2. Hasil Prediksi Model MobileNetV2
- Lampiran 3. Hasil Prediksi Model MobileNetV2 Fine Tuning
- Lampiran 4. Hasil Prediksi Model EfficientNetB0
- Lampiran 5. Dokumentasi Penelitian
- Lampiran 6. Tampilan Identik Sarang Setiap Jenis Lebah
- Lampiran 7. Link Google Drive Dataset dan Github Repository
- Lampiran 8. Plagiarism

## I. PENDAHULUAN

### 1.1 Latar Belakang

Lebah madu merupakan salah satu *family Apidae* dalam *subfamily Apinae* yang berperan penting dalam mempertahankan keberlanjutan ekosistem maupun keanekaragaman hayati. Lebah madu berperan dalam membantu proses penyebukan tanaman atau reproduksi tanaman. Hal ini menunjukkan bahwa populasi lebah juga akan berdampak pada stabilitas produksi pangan dan keanekaragaman hayati, sehingga perlindungan terhadap lebah madu menjadi salah satu prioritas penting dalam upaya konservasi pertanian berkelanjutan[1] . Lebah madu pada umumnya terbagi menjadi lebah sengat (*Apis sp.*) dan lebah tanpa sengat (*Meliponini sp.*). Jenis lebah ini juga akan membedakan kandungan pada madu yang dihasilkan. Menurut *research* disebutkan bahwa kandungan antoksidan pada madu dari lebah tanpa sengat atau *stingless bee* lebih tinggi dibandingkan madu lebah apis. Sehingga madu lebah tanpa sengat tergolong sebagai salah satu bahan pangan alami yang paling kompleks [2].

Di Indonesia, lebah madu ini tidak hanya berperan sebagai penghasil madu alami saja, melainkan juga membentuk komoditas penting seperti *royal jelly* dan *propolis*. Dikutip dari *Journal of Food Science and Technology* bahwa *royal jelly* adalah makanan khusus bagi ratu lebah yang memiliki banyak manfaat kesehatan dan sering digunakan dalam industri kosmetik maupun kesehatan [3]. Adapun *propolis* yang dikenal sebagai lem lebah ini memiliki nilai tinggi. Dikutip dari *Journal of Pharmaceutical Sciences and* bahwa *propolis* yang dihasilkan lebah dari resin tanaman memiliki sifat *research anti-mikroba* dan *anti-inflamasi* sehingga bermanfaat dalam pengobatan tradisional dan produk kesehatan [4]. *Research* juga menunjukkan bahwa permintaan domestik terhadap produk olahan lebah madu meningkat, namun hasil produksi madu di Indonesia belum memenuhi permintaan, sehingga Indonesia masih menjadi salah satu negara importir madu di dunia. Maka dari itu, kondisi ini dapat menjadi peluang ekonomi domestik bagi peternak lebah

lokal untuk dapat menghasilkan beragam produk olahan bermilai tinggi dengan peningkatan kualitas madu yang baik pula. Dalam upaya peningkatan produksi lebah madu oleh peternak lebah, tentunya terdapat tantangan dalam pengelolaan lebah terutama dalam pemanfaatan teknologi dan masih terbatasnya keterampilan peternak lebah dalam pengelolaan tiap *spesies* lebah tanpa sengat yang memiliki karakteristik visual serupa dari total sekitar 500-600 spesies yang diakui, namun memiliki potensi hasil madu yang berbeda-beda.

Menurut penelitian sebelumnya saat ini perkembangan teknologi seperti sistem otomatisasi dan pemodelan klasifikasi sudah cukup banyak dikembangkan. Penelitian terdahulu yang mengembangkan pemodelan klasifikasi lebah pada tiga jenis lebah menunjukkan akurasi yang cukup tinggi untuk dapat membedakan karakteristik visual lebah dari tiga jenis yang diambil dan dilatih. Sementara penelitian pemodelan klasifikasi lainnya seperti pada klasifikasi *spesies* kupu-kupu dan klasifikasi penyakit pada tanaman daun tembakau juga menunjukkan bahwa model klasifikasi yang dilatih dapat mencapai akurasi yang cukup tinggi dalam membantu kesulitan klasifikasi secara manual yang menuntut detail visual yang serupa seperti pada objek *spesies* lebah madu. Namun masih adanya keterbatasan penelitian sebelumnya dalam variasi objek data lebah, selain itu perlu adanya fokus pada proses teknik *pre-processing* dan *augmentasi* data yang dapat mempengaruhi hasil akurasi model yang dilatih pada dataset.

Maka dari itu, berdasarkan uraian sebelumnya penelitian ini akan mengembangkan model klasifikasi spesies lebah madu tanpa sengat atau *stingless bee* (*Meliponini sp.*). Pemodelan ini dikembangkan untuk mendukung peternak lebah agar pengelolaan klasifikasi *spesies* lebah lebih optimal dan terarah. Penelitian ini akan berfokus pada proses implementasi model klasifikasi *spesies* lebah madu dengan arsitektur *deep learning*. Adapun data citra gambar lebah madu tanpa sengat yang digunakan dalam penelitian ini diambil dari Lembah Suhita sebagai tempat eduwisata lebah madu. Pengembangan model klasifikasi ini menggunakan arsitektur *Convolutional Neural Network (CNN)*. Dimana arsitektur *CNN* terbukti sebagai metode yang efisien dalam klasifikasi gambar karena mampu mengenali pola dalam data citra tanpa memerlukan ekstraksi fitur manual. Penelitian menunjukkan bahwa *CNN* memiliki akurasi yang cukup tinggi dan

efisiensi komputasi yang baik [5]. Teknik *augmentasi* data pada tahap *preprocessing* juga akan dioptimalkan, karena pada penelitian sebelumnya terbukti bahwa teknik augmentasi dapat meningkatkan hasil akurasi model [6].

Penelitian ini diharapkan dapat memberikan kontribusi terhadap upaya peningkatan ternak lebah madu yang diangkat terkait klasifikasi *spesies* lebah madu tanpa sengat yang berpotensi untuk dikembangkan. Dengan pemodelan klasifikasi ini, diharapkan pengelompokan spesies dengan penggunaan model ini dapat meningkatkan fokus optimalisasi pengelolaan peternak pada spesies lebah madu tanpa sengat. Sehingga, peternak lebah dapat mengoptimalkan kualitas hasil dan diversifikasi hasil lebah madu tanpa sengat, yang terbukti memiliki potensi kandungan nutrisi, antioksidan, dan nilai jual yang lebih tinggi dibanding lebah madu sengat (*Apis sp.*). Selain itu, penelitian ini juga diharapkan dapat memberikan wawasan baru mengenai *spesies* lebah madu tanpa sengat dan pengembangan keilmuan sejenis terkait pemodelan klasifikasi citra menggunakan *deep learning*. Dengan demikian, hasil penelitian ini tidak hanya bermanfaat bagi praktisi di lapangan, tetapi juga bagi akademisi dan penelitian selanjutnya.

## 1.2 Rumusan Masalah

Adapun rumusan masalah yang diangkat pada penelitian ini adalah sebagai berikut:

1. Bagaimana mengembangkan sistem kecerdasan buatan melalui model *deep learning* CNN untuk mengklasifikasi citra lebah madu tanpa sengat (*stingless*)?
2. Bagaimana mengimplementasikan model CNN MobileNetV2 dan EfficientNetB0 menggunakan teknik augmentasi pada citra lebah madu tanpa sengat (*stingless*)?
3. Bagaimana pengaruh hyperparameter pada performa model CNN MobileNetV2 dan EfficientNetB0 untuk klasifikasi citra lebah madu tanpa sengat (*stingless*)?

## 1.3 Tujuan Penelitian

Adapun tujuan penelitian yang diangkat pada penelitian ini adalah sebagai berikut:

1. Mengembangkan sistem kecerdasan buatan dengan model deep learning *Convolutional Neural Network* (CNN) untuk klasifikasi citra lebah madu tanpa sengat (*stingless*).
2. Mengimplementasikan teknik augmentasi pada model *Convolutional Neural Network* (CNN) MobileNetV2 dan EfficientNetB0 untuk klasifikasi citra lebah madu tanpa sengat (*stingless*).
3. Mengevaluasi performa model *Convolutional Neural Network* (CNN) MobileNetV2 dan EfficientNetB0 dalam klasifikasi citra lebah madu tanpa sengat (*stingless*) dari aspek hyperparameter.

#### **1.4 Manfaat Penelitian**

Adapun manfaat yang diharapkan dari hasil pada penelitian ini adalah sebagai berikut:

1. Secara Akademis, penelitian ini dapat meningkatkan pengalaman dalam pengembangan model *deep learning* menggunakan *Convolutional Neural Network* (CNN) model MobileNetV2 dan EfficientNetB0 untuk klasifikasi citra *spesies* lebah madu tanpa sengat (*stingless*) bagi peneliti.
2. Secara Praktis, penelitian ini memberikan solusi praktis yang cukup efisien dalam klasifikasi *spesies* lebah madu tanpa sengat yang dapat digunakan dalam industri peternakan lebah, penelitian biodiversitas, dan mendukung penerapan teknologi modern di industri terkait, sehingga membuka peluang untuk peningkatan produktivitas di bidang terkait serta mendukung optimalisasi keberlanjutan ekosistem melalui peningkatan populasi lebah.
3. Secara Teknis, penelitian ini menunjang dalam penyediaan dataset yang aktual dari empat *spesies* citra lebah madu tanpa sengat di Lembah Suhita yang dapat digunakan sebagai referensi dan memperkaya dataset untuk penelitian sejenis di bidang pengolahan data yang relevan terkait citra digital dan pemodelan klasifikasi citra lebah.

#### **1.5 Batasan Masalah**

Adapun penelitian ini, dibatasi oleh beberapa batasan masalah sebagai berikut:

1. Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) MobileNetV2 dan EfficientNetB0, dan tidak membandingkan hasil dengan metode lain.
2. Penelitian ini hanya mengimplementasikan model klasifikasi CNN MobileNetV2 dan EfficientNetB0 dengan mengevaluasi hasil performa dan tidak mencakup tahap *deployment* sistem.
3. Dataset yang digunakan dalam penelitian ini hanya mencakup empat kelas *spesies* lebah madu tanpa sengat (*stingless*) dari Lembah Suhita, dan tidak mencakup semua *spesies* lebah madu tanpa sengat yang ada di dunia.
4. Variabel lingkungan dalam pengambilan data citra hanya diambil pada pencahayaan alami di siang hari dan pencahayaan buatan.

## **1.6 Sistematika Penulisan**

Adapun struktur penulisan yang diterapkan pada penelitian ini adalah sebagai berikut:

### BAB I : PENDAHULUAN

Dalam bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dari pengembangan model klasifikasi citra lebah madu tanpa sengat, dan sistematika penulisan laporan.

### BAB II : TINJAUAN PUSTAKA

Dalam bab ini berisi tentang dasar-dasar teoritis pendukung dari berbagai sumber seperti buku, jurnal, atau paper terkait topik penelitian yang diangkat, *tools*, cara kerja implementasi, metode pengembangan, dan penelitian terdahulu yang relevan terkait judul penelitian sebagai literatur bacaan pengembangan model.

### BAB III : METODE PENELITIAN

Dalam bab ini berisi tentang langkah-langkah sistematis dan logis yang memuat waktu penelitian, tempat penelitian, alat penelitian, bahan penelitian, dan tahapan penelitian dalam implementasi pengembangan model klasifikasi citra lebah madu tanpa sengat dengan arsitektur *Convolutional Neural Network (CNN)*.

**BAB IV : HASIL DAN PEMBAHASAN**

Dalam bab ini berisi pembahasan penelitian dari tahap *planning*, pengumpulan data, *preprocessing* data, augmentasi data, implementasi model *CNN*, training model, pengujian, evaluasi model, dan temuan terkait hasil dan penelitian model klasifikasi *spesies* lebah madu tanpa sengat (*stingless*) yang dilatih.

**BAB V : PENUTUP**

Dalam bab ini berisi kesimpulan berdasarkan hasil penelitian terkait pengembangan model klasifikasi citra lebah madu tanpa sengat (*stingless*) serta berisi saran untuk pengembangan penelitian terkait atau serupa.

**DAFTAR PUSTAKA**

**LAMPIRAN**

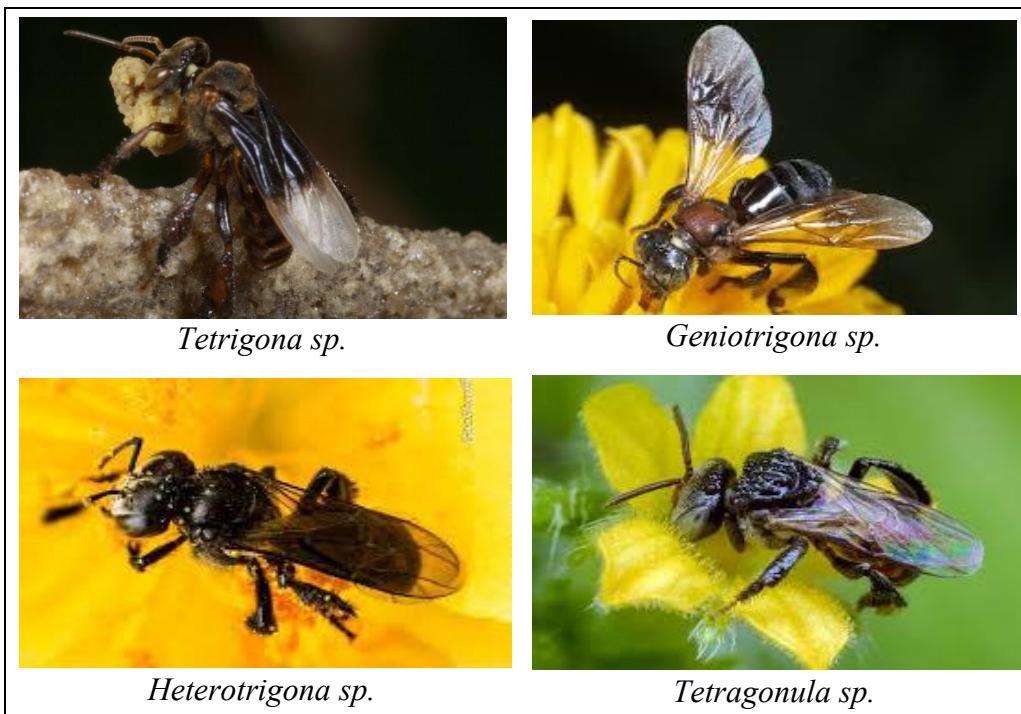
## II. TINJAUAN PUSTAKA

### 2.1 Dasar Teori

#### 2.1.1 *Spesies Lebah Madu*

Lebah madu merupakan salah satu *family Apidae* dalam *subfamily Apinae* yang berperan penting dalam mempertahankan keberlanjutan ekosistem maupun keanekaragaman hayati. Lebah madu hidup dalam koloni yang terdiri dari tiga peranan yaitu ratu (*queen*) sebagai lebah betina yang reproduktif, pekerja (*worker*) sebagai lebah betina yang tidak dapat bereproduksi, dan *drone* sebagai lebah jantan yang bertugas mengawini ratu dari koloni lain. Lebah madu juga terbagi menjadi 2 jenis berdasarkan sengat, yaitu lebah madu dengan sengat (*Apis sp.*) dan lebah madu tanpa sengat (*Meliponini sp.*) atau dikenal *stingless bee*. Setiap *spesies* lebah ini memiliki karakteristik unik dalam hal morfologi, perilaku, dan adaptasi terhadap lingkungan masing-masing. Diluar *spesies* lebah yang beragam, lebah madu sama-sama memiliki peran penting secara tidak langsung dalam pertanian dan produksi pangan melalui proses penyerbukan, serta menghasilkan produk bernilai ekonomi seperti madu, *royal jelly*, dan *propolis*. Namun, madu yang dihasilkan oleh tiap jenis lebah yang berbeda akan memiliki sifat fisik dan kimia yang berbeda-beda pula [7]. Seperti *spesies* lebah madu genus *apis* yang banyak dibudidayakan yaitu *apis mellifera* (Eropa) dan *apis cerana* (Asia) menghasilkan madu, lilin, propolis, dan berkontribusi pada 75% penyerbukan tanaman pangan global [8]. Sementara jenis lebah madu tanpa sengat (*stingless bee*) terdiri dari beberapa genus seperti *trigona* sp., *tetrigona* sp. dan lainnya. Penelitian sebelumnya menyebutkan bahwa *spesies* lebah madu tanpa sengat seperti *Tetragonula carbonaria* dan *Melipona beecheii* menghasilkan madu bernutrisi tinggi dengan kadar air lebih tinggi dibanding madu lebah *apis*, serta berperan dalam penyerbukan tanaman hutan dan agrofestari [9]. Maka dari itu, potensi akan hasil madu dari lebah madu tanpa sengat (*stingless bee*) bernilai lebih tinggi daripada lebah madu genus *apis*.

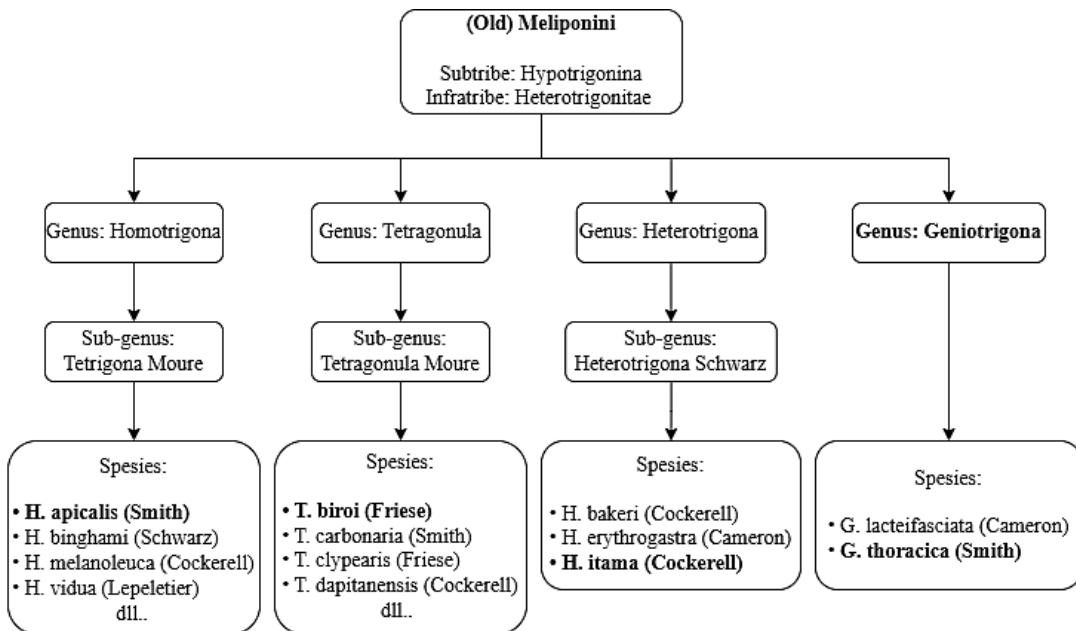
Adapun pada penelitian ini akan menggunakan data 4 jenis lebah madu tanpa sengat (*stingless*) yaitu *tetrigona spp.*, *heterotrigona sp.*, *geniotrigona sp.*, dan *tetragonula spp.*. Masing-masing jenis lebah ini memiliki karakteristik morfologi tertentu, seperti warna, corak, abnomen lebah, sayap, ukuran, dan aspek lainnya yang mungkin sulit dibedakan. Berikut adalah gambar keempat jenis atau *spesies* lebah madu *stingless* yang akan dijadikan objek pada penelitian ini:



(Sumber: Flickr.com)

Gambar 2. 1. Lebah Madu Stingless

Berdasarkan Gambar 2.1. sebagai gambar lebah madu stingless dari 4 jenis lebah yang akan digunakan sebagai *object* pemodelan klasifikasi jenis lebah dalam penelitian ini. Secara umum keempat lebah tersebut memiliki ukuran tubuh dari 3mm sampai dengan 9mm. Secara visual keempat lebah memiliki perbedaan yang mungkin tidak dapat ditangkap langsung karakteristik visualnya terutama oleh orang lain di bidang lain atau masyarakat umum. Perbedaan genus secara spesifik akan membuat tiap *spesies* lebah memiliki ciri tersendiri. Berikut ini akan dijelaskan terkait taksonomi dari empat jenis lebah yang digunakan dimana empat jenis ini berasal dari genus yang berbeda-beda.



(Sumber: Engel MS, Rasmussen C, Ayala R, de Oliveira FF, Zoo Keys 2023)

Gambar 2. 2. Pohon Kelas Lebah yang Diteliti

Berdasarkan sumber artikel internasional *Zoo Keys* tahun 2023 yang menyajikan *classification* dan *biology* dari *stingless bee* diperoleh bahwa lebah tanpa sengat (*stingless*) dalam *sub family apidae* yang digunakan dalam penelitian ini tergolong dalam tribe Meliponini. Dalam tribe ini memiliki beberapa genus yang mana empat jenis lebah yang digunakan dalam penelitian ini secara spesifik dikelompokkan di bawah infratribe Heterotrigonitae. Berdasarkan Gambar 2.2 di atas empat genus data lebah yang digunakan dalam penelitian ini terdiri atas *Homotrigona*, *Tetragonula*, *Heterotrigona*, dan *Geniotrigona*. Dalam klasifikasi ini, setiap genus diambil satu jenis atau spesies dari sub-genus *Tetragonula* yang digunakan yaitu *T. Biroi*, sub-genus *Heterotrigona* yaitu *H. Itama*, sub-genus *Geniotrigona* mencakup spesies *G. Thoracica*, dan terakhir ada sub-genus *Tetrigona* sebagai sub-genus di bawah *Homotrigona*, yang mana merupakan asal spesies lebah *H. apicalis*. Namun, dari beberapa penelitian acuan kelas yang lebih *modern* tribe meliponini terbaru, klasifikasi saat ini telah memperbarui pohon kelas tersebut, di mana beberapa sub-genus telah dinaikkan statusnya menjadi genus penuh yaitu pada *Homotrigona* dan *Tetrigona* yang sekarang diakui sebagai dua genus yang sepenuhnya terpisah dan berbeda.

### 2.1.2 Artificial Intelligence

*Artificial Intelligence* (AI) atau Kecerdasan Buatan adalah salah satu cabang ilmu komputer yang diterapkan dalam pengembangan sistem atau komputer cerdas yang dapat melakukan tugas layaknya seperti dari kecerdasan manusia. AI menerapkan prinsip algoritma dan model matematika untuk belajar dari data sehingga dapat mengenali pola data dan membentuk sistem keputusan yang cerdas [10]. Definisi AI sebagai sistem cerdas dapat dijelaskan dalam empat sudut pandang karakteristik kinerja dan rasionalitasnya seperti pada tabel berikut.

Tabel 2. 1. Definisi AI dalam Sudut Pandang Karakteristik

<b><i>Thinking Humanly</i></b> Sistem Berpikir Layaknya Manusia (implementasi bahasa pemrograman yang dapat berpikir seperti manusia)	<b><i>Acting Humanly</i></b> Sistem Yang Bertindak Seperti Manusia (implementasi komputasi dan test yang setara bahkan dapat melebihi manusia)
<b><i>Thinking Rationally</i></b> Sistem Berpikir Rasional (implementasi sistem yang handal dapat memutuskan tugas)	<b><i>Acting Rationally</i></b> Sistem Bertindak Rasional (implementasi sistem persepsi dan dapat menyesuaikan dengan perubahan)

Pada Tabel 2.1. mendefinisikan AI dalam sudut pandang rasional dan selayaknya kinerja manusia. Proses kinerja layaknya manusia yang dimaksud yaitu dimana AI melibatkan cara berpikir dan bertindak selayaknya manusia dengan penalaran, *instinct*, dan variabel lain diluar data yang ada secara rasional. Sementara secara rasional AI berpikir dan bertindak rasional jika kinerja sistem mencapai hasil yang terbaik. AI dapat mempelajari sejumlah data besar untuk menghasilkan sebuah sistem atau komputer cerdas melalui 3 aspek keterampilan kognitif berupa penalaran (*reasoning*), pembelajaran (*learning*), dan kreatifitas (*creativity*) sehingga sistem cerdas dapat menganalisis, mendiagnosa, mengidentifikasi, memprediksi, dan mengambil tindakan [11]. Secara umum diharapkan tujuan pengembangan dan penerapan AI meliputi [12] :

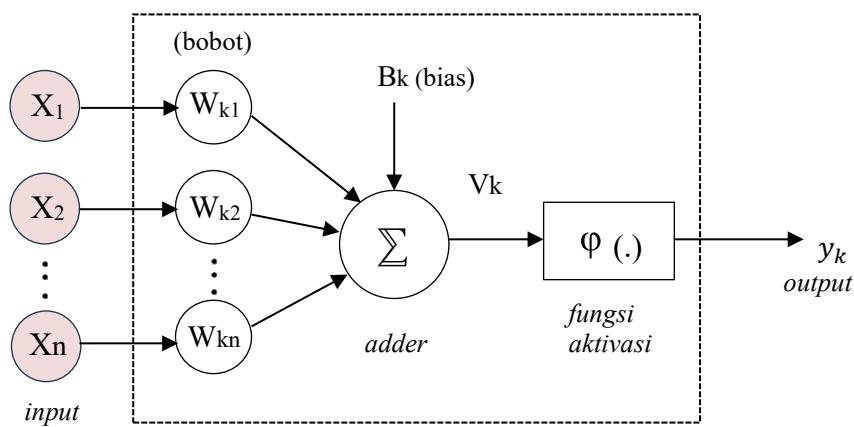
1. AI dapat diaplikasikan pada program tertentu seperti robot yang dapat membantu kegiatan manusia.
2. AI dapat membantu dalam pemecahan masalah yang kompleks atau rumit, seperti dengan perhitungan cepat dengan sistem kalkulator cerdas.
3. AI dapat meningkatkan kinerja sistem atau komputer menjadi lebih cerdas dari sebelumnya.

*Artificial Intelligence (AI)* atau Kecerdasan Buatan terdiri dari beberapa cabang ilmu pengetahuan mencakup *Machine Learning*, *Natural Language Processing*, *Computer Vision*, *Expert System*, *Robotics*, *Knowledge-Based Systems and Reasoning*, dan *AI-Generative*. Dalam pemahamannya AI memiliki 3 konsep dasar mencakup *Machine Learning*, *Deep learning*, dan *Artificial Neural Network* atau Jaringan Saraf Tiruan. Saat ini AI berkembang dalam berbagai bidang kehidupan dan telah banyak diimplementasikan dalam memudahkan permasalahan yang rumit seperti bidang berikut ini:

1. Kesehatan, seperti dalam diagnosis penyakit melalui analisis citra medis pada *CT-Scan*, *X-Ray*, dan prediksi lainnya.
2. Pendidikan, seperti *adaptive learning* pada *platform* pembelajaran sesuai kemampuan siswa, dan *generatif-AI* untuk fitur tanya jawab otomatis.
3. Industri, seperti pada proses *quality control* dengan *computer vision* untuk inspeksi cacat produk di proses produksi.
4. Otomatisasi, seperti pada teknologi *chatbot* layanan dalam tanya jawab pengguna dan *smart home* untuk sistem otomatis.
5. Transportasi, seperti optimasi rute pada penggunaan Waze atau Google Maps dalam rekomendasi jalur tercepat.
6. *E-Commerce* dan Bisnis, seperti pada rekomendasi produk personalisasi sesuai kebiasaan pengguna dan deteksi penipuan transaksi.
7. Teknologi Informasi dan Komunikasi, seperti penggunaan *voice assistant* pada *Google Assistant* untuk perintah suara.
8. Pemerintahan dan Administrasi, seperti pada *management* pengelolaan untuk meningkatkan layanan publik, analisis, dan pengambilan kebijakan.

### 2.1.3 Artificial Neural Network

*Artificial Neural Network* (ANN) atau disebut Jaringan Saraf Tiruan (JST) adalah salah satu model pengembang dari *Machine Learning* yang terinspirasi dari cara kerja jaringan saraf biologis pada otak manusia untuk memproses pembelajaran pada data model. Jaringan saraf ini tersusun dari neuron-neuron buatan sebagai unit untuk memproses informasi data berupa pengetahuan atau *knowledges* dari proses pelatihan atau komputasi secara paralel dalam skala besar. Berikut adalah gambaran ilustrasi neuron pada Jaringan Saraf Tiruan.



(Sumber: Setiawan Wahyu, 2020)

Gambar 2. 3. Ilustrasi Neuron pada Jaringan Saraf Tiruan

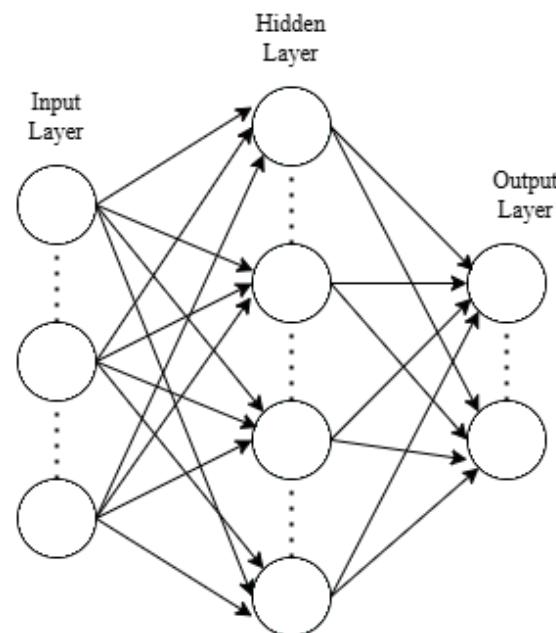
Serupa dengan halnya neuron biologis pada otak manusia yang saling berhubungan satu sama lain, jaringan saraf tiruan (JST) juga memiliki neuron-neuron yang saling berhubungan satu sama lain dalam berbagai lapisan jaringan. Pada Gambar 2.3. mengilustrasikan bahwa JST tersusun dari input, bobot, bias, fungsi penjumlahan, fungsi aktivasi, dan output akhir. Berikut penjelasan terkait komponen pada Gambar 2.3. Ilustrasi Neuron pada Jaringan Saraf Tiruan.

1.  $X_1, X_2, \dots, X_n$  (Input) adalah masukan atau fitur yang diberikan ke neuron. Setiap nilai  $X_n$  mewakili satu atribut atau variabel dalam dataset yang digunakan untuk pelatihan atau prediksi.
2.  $W_{k1}, W_{k2}, \dots, W_{kn}$  (Bobot) adalah ukuran yang menunjukkan seberapa besar pengaruh masing-masing input terhadap output. Bobot ini akan dioptimalkan selama proses pelatihan model JST.

3.  $B_k$  (Bias) adalah nilai tambahan yang membantu menyesuaikan output neuron. Bias memungkinkan model untuk melakukan translasi fungsi aktivasi sehingga jaringan dapat belajar lebih *fleksibel*.
4.  $V_k$  (Adder) adalah fungsi yang digunakan untuk penjumlahan berbobot yang menghitung nilai total dari input yang diberikan ke neuron berdasarkan persamaan berikut.

$$V_k = \sum_{i=1}^n W_{ki} X_i + B_k$$

5.  $\varphi$  (Fungsi Aktivasi) adalah fungsi yang digunakan untuk mengambil nilai hasil perhitungan dari adder ( $V_k$ ) untuk menghasilkan output akhir prediksi model. Fungsi aktivasi yang digunakan dapat berupa Sigmoid, ReLU, Tanh, Softmax, dan lainnya.
6.  $Y_k$  (Output) adalah hasil dari neuron setelah dikali fungsi aktivasi yang dapat digunakan sebagai masukan ke neuron lain dalam lapisan dalam berikutnya, jika hasil belum optimal atau sebagai hasil akhir jika sebagai lapisan output.



(Sumber: Jamaaluddin dan Sulistiowati,I , 2021)

Gambar 2. 4. Contoh Arsitektur Jaringan Saraf Tiruan

Dalam komputasi pemodelan jaringan saraf tiruan memiliki susunan lapisan (*layers*) yang terdiri atas *input layer*, *hidden layer*, dan *output layer*. Arsitektur ini digambarkan pada Gambar 2.4. Berdasarkan gambar tersebut pada *layer input* JST akan menerima data masukkan dari luar dan jumlah input disesuaikan dengan fitur atau atribut yang akan digunakan karena setiap neuron input akan mewakili fitur atau atribut data masukkan. Pada *layers hidden* neuron-neuron akan melakukan transformasi *non-linier* terhadap data masukkan untuk mengekstrasi fitur-fitur yang lebih atau semakin kompleks. Lalu pada *layer output* akan menghasilkan prediksi berdasarkan hasil ekstraksi data pada layer-layer sebelumnya.

Jaringan Saraf Tiruan (JST) banyak diterapkan karena memiliki keunggulan dalam kemampuan belajar dan beradaptasi dari berbagai tipe data, mengenali pola kompleks, menangani hubungan *non-linear*. Adapun jenis-jenis arsitektur jaringan saraf tiruan yang sering digunakan antara lain [11] :

- 1 *Perceptron* adalah salah satu jenis arsitektur JST yang hanya terdiri dari satu neuron input, tidak memiliki *hidden layer*, dan menerapkan fungsi aktivasi untuk menghasilkan output biner.
- 2 *Feedforward Neural Network* (FNN) adalah arsitektur JST yang memiliki prinsip dimana informasi mengalir maju tanpa adanya siklus mundur. Tiap neuron saling terhubung dari lapisan (layer) input, ke *hidden layer*, lalu ke layer output yang melibatkan variabel bobot dan fungsi aktivasi. FNN ini sangat baik dalam mempelajari pola-pola yang kompleks dalam data.
- 3 *Convolutional Neural Network* (CNN) adalah arsitektur JST yang dirancang khusus untuk pengolahan data spasial seperti citra atau gambar. CNN mampu bekerja efisien dalam ekstraksi fitur penting dari input citra gambar. CNN terdiri dari lapisan konvolusi, pooling, fungsi aktivasi, dan *fully connected*.
- 4 *Recurrent Neural Networks* (RNN) adalah arsitektur JST yang dilatih untuk memproses dan mengubah input data berurutan menjadi output data berurutan tertentu. RNN memiliki sambungan siklik seperti cincin, yang memungkinkan informasi untuk mengalir mundur ke dalam jaringan.

#### 2.1.4 Deep Learning

*Deep Learning* adalah bagian dari *Machine Learning* yang termasuk dalam pengembangan *Artificial Neural Network* (ANN). Istilah ‘*deep*’ pada *deep learning* merujuk pada proses pembelajaran dari banyak layer, dimulai dari layer awal input yang sederhana hingga layer dalam yang lebih kompleks. Metode *deep learning* sering ditemukan pada studi kasus seperti klasifikasi, *clustering*, *segmentasi*, maupun *recognition* pada data tidak teratur seperti teks, suara, ataupun citra [13]. Semakin banyak model belajar terhadap data yang diekstraksi, maka akan semakin akurat pula akurasi model yang dihasilkan.

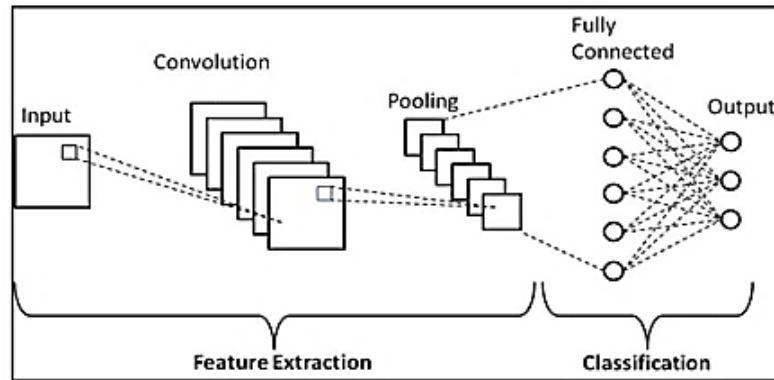
*Deep learning* dapat diimplementasikan pada tiga skenario pembelajaran mesin mencakup *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Pembelajaran *supervised* dilakukan model pada data berlabel, *unsupervised* dilakukan model pada data tidak berlabel, dan *reinforcement* dilakukan melalui uji *trial* dan *error*, dimana model mendapatkan *reward* atau *punishment* berdasarkan hasil. *Deep learning* memiliki cakupan model yang luas dengan ragam model pembelajaran yang saat ini telah menjadi pondasi penting dalam perkembangan teknologi inovasi di berbagai bidang, seperti pada *platform media sosial* untuk *automatic text reply*, pada *browser* pencarian untuk fitur pencarian gambar dan penerjemah otomatis, pada *smartphone* untuk fitur pengenalan suara dan deteksi wajah, pada teknologi navigasi, dan pada bidang kesehatan seperti diagnosis medis melalui teknologi *MRI* atau *rontgen* [14].

Keunggulan utama *deep learning* terletak pada kemampuan pada arsitektur jaringan saraf (*neural network*) untuk menyesuaikan jutaan parameter bobot atau *weight* secara dinamis menggunakan algoritma *backpropagation*. Hasil prediksi yang dihasilkan akan dibandingkan dengan nilai target sebenarnya melalui *loss function*, lalu kesalahan ini menjadi umpan balik untuk memperbaiki bobot setiap lapisan (*layer*) [15], [16]. *Deep learning* terdiri dari beberapa komponen inti untuk membangun dan melatih model, meliputi lapisan (*layer*) *convolutional* untuk ekstraksi fitur citra, *dense layer* untuk koneksi antar neuron, dan *activation layer* untuk memberikan *non linearitas*, bobot (*weight*) sebagai parameter yang disesuaikan selama pelatihan, fungsi aktivasi yang menentukan *output* neuron, fungsi loss (*loss function*) untuk mengukur kesalahan prediksi, *optimizer* seperti

Adam, SGD, dan lainnya dalam memperbarui bobot untuk meminimalisirkan kesalahan saat proses pelatihan, serta teknik regularisasi seperti *dropout* untuk mencegah *overfitting* dengan membatasi kompleksitas model. Semua komponen ini bekerja secara terintegrasi untuk menghasilkan model yang akurat [16].

### 2.1.5 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) adalah salah satu jenis arsitektur dalam *deep learning* yang biasa digunakan untuk tugas seperti klasifikasi gambar, prediksi atau pengenalan pola. CNN sebagai arsitektur *deep learning* dengan jaringan saraf memiliki prinsip kerja seperti neuron pada otak manusia. Pada dasarnya CNN mengirimkan data dari satu lapisan (layer) ke lapisan lebih dalam yang mana setiap lapisan akan memproses informasi dari banyak neuron. Layer-layer ini memungkinkan model CNN belajar dari deteksi pengenalan pola tingkat rendah hingga tingkat tinggi, seperti model klasifikasi lebah yang membutuhkan presisi dalam membedakan morfologinya. Dalam implementasinya neuron dapat memiliki input yang sama namun nilai bobot berbeda [13]. Nilai bobot ini ditentukan dan dapat diperbarui selama proses pelatihan untuk penyesuaian dan pembelajaran terhadap data yang lebih baik. Arsitektur CNN memiliki kemampuan menghasilkan fitur-fitur yang relevan berdasarkan input citra yang diberikan. Lapisan (layer) pada arsitektur CNN terbagi menjadi dua berdasarkan fungsi, yaitu layer ekstraksi fitur dan layer klasifikasi. Berdasarkan Gambar 2.5. Layer ekstraksi fitur terdiri dari layer konvolusional, *pooling layer*, dan *Rectified Linier Unit* (ReLU), sementara layer klasifikasi terdiri dari *fully connected layer*, *dropout*, dan *softmax (fungsi aktivasi)* [13]. Pada *feature extraction* input citra akan diekstrasi mulai dari pola sederhana seperti tepi, warna, atau tekstur hingga pola kompleks seperti bentuk atau bagian objek spesifik. Sementara pada *fully connected layer* fitur yang diekstraksi akan diintegrasikan untuk menghasilkan output klasifikasi berdasarkan pola yang telah dipelajari sebelumnya.



(Sumber: Bismi.W dan Qomarudin.M, 2023)

Gambar 2. 5. Arsitektur Layers pada Proses CNN

Penggunaan arsitektur CNN memiliki beberapa keunggulan seperti fitur pembagian bobot, yang mengurangi jumlah parameter jaringan yang dapat dilatih dan membantu jaringan untuk meningkatkan generalisasi dan menghindari *overfitting*. Selain itu, CNN mempelajari kedua bagian fungsi layer bersamaan sehingga keluaran model menjadi terorganisir dan juga cocok digunakan dalam implementasi jaringan skala besar. Dalam implementasi arsitektur CNN sering kali terjadi *underfitting* dan *overfitting*, maka dari itu terdapat beberapa teknik regulasi yang dapat dilakukan untuk menghindari hal tersebut seperti teknik *dropout*, teknik augmentasi data, dan *batch normalization*. Model dalam arsitektur CNN saat ini sudah banyak dikembangkan, menjadi dasar dalam pengembangan tugas seperti klasifikasi citra, deteksi objek, segmentasi, dan pengolahan citra medis. Adapun contoh model arsitektur CNN populer seperti AlexNet, VGG, dan ResNet [17]. Berikut penjelasan setiap layer dalam susunan arsitektur layer CNN pada pemodelan klasifikasi:

### 1. Layer Input

Layer Input merupakan layer awal dimana data mentah dimasukkan ke dalam jaringan model. Pada konteks *deep learning* pemodelan data yang dimasukkan umumnya adalah citra atau gambar digital. Layer ini tidak melakukan proses pembelajaran atau operasi komputasi, melainkan hanya berfungsi sebagai penampung data yang mendefinisikan bentuk dari data input. Bentuk ini biasanya

direpresentasikan dalam format tiga dimensi yaitu tinggi citra, lebar citra, dan jumlah channel warna, seperti channel 3 untuk citra berwarna (RGB) atau 1 untuk citra *grayscale* [15].

## 2. Layer Convolution

Layer konvolusi merupakan layer penyusun utama dari sebuah model *Convolutional Neural Network* (CNN). Tujuannya adalah untuk mengekstraksi fitur-fitur dari citra input. Layer ini bekerja dengan menggeser sebuah filter atau *kernel*, yaitu sebuah matriks bobot berukuran kecil, ke seluruh area citra. Proses pergeseran ini disebut sebagai operasi konvolusi. Setiap filter secara otomatis belajar untuk mendeteksi fitur spesifik, seperti tepi, sudut, tekstur, atau bahkan bagian objek yang lebih kompleks pada layer yang lebih dalam. Hasil dari operasi satu filter pada citra adalah sebuah *feature map* atau peta fitur, yang menyorot area di mana fitur tersebut terdeteksi. Parameter utama pada layer ini meliputi jumlah filter, ukuran kernel, *stride*, dan *padding* atau penambahan piksel di sekeliling citra [18]. Proses belajar dalam ekstrasi fitur otomatis ini yang membuat CNN sesuai dan akurat untuk tugas-tugas pengenalan visual seperti citra [19].

## 3. Layer Pooling

Layer Pooling adalah layer yang berada setelah layer konvolusi dan bertujuan untuk mengurangi dimensi spasial berupa tinggi dan lebar dari *feature map*. Proses ini memiliki manfaat untuk mengurangi jumlah parameter dan komputasi dalam jaringan sehingga lebih efisien. Selain itu, layer ini akan membuat representasi fitur menjadi lebih robust terhadap variasi posisi dan rotasi kecil pada citra. Terdapat dua jenis pooling yang umum digunakan dalam pemodelan yaitu:

- a. Max Pooling, merupakan layer pooling yang cara kerjanya mengambil nilai piksel maksimum dari setiap jendela pada *feature map*, yang mana layer ini efektif untuk mempertahankan fitur yang paling menonjol.
- b. Average Pooling, merupakan layar polling yang prinsipnya mengambil nilai rata-rata dari setiap jendela, dengan mengurangi resolusi sehingga membantu jaringan untuk fokus pada keberadaan fitur daripada lokasi persisnya [15].

## 4. *Fully Connected Layer* (Dense)

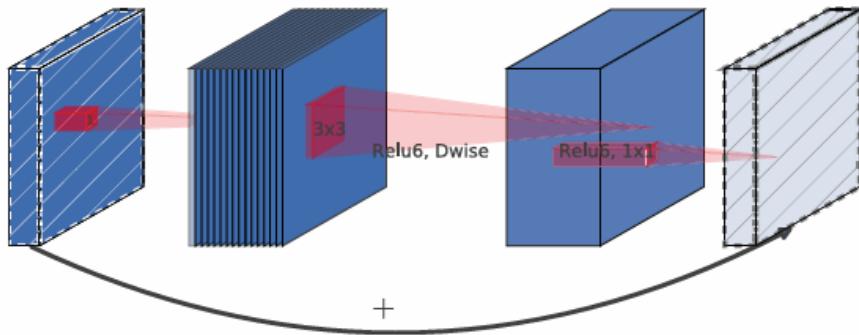
Selanjutnya setalah fitur-fitur penting citra diekstraksi oleh layer konvolusi dan disederhanakan oleh layer pooling, *feature map* akhir yang berbentuk 2D akan menjadi sebuah vektor tunggal atau flatten. Vektor ini kemudian dimasukkan ke dalam satu atau lebih *Fully Connected* atau *Dense layer*. Dalam layer ini, setiap neuron terhubung ke semua neuron dari layer sebelumnya. Tujuannya adalah untuk melakukan klasifikasi berdasarkan fitur tingkat tinggi yang telah diekstraksi. Layer akan belajar kombinasi non-linear dari fitur-fitur tersebut untuk dapat membedakan satu kelas dengan kelas lainnya. Saat layer konvolusi mendeteksi fitur mata, sayap, antene, dan lainnya maka layer dense akan bertugas mempelajari untuk mengombinasikan fitur-fitur ini untuk mengklasifikasikannya sebagai objek lebah [18].

## 5. Layer Output

Layer Output adalah layer terakhir dalam jaringan yang menghasilkan prediksi akhir. Jumlah neuron pada layer ini biasanya sama dengan jumlah kelas yang akan diprediksi. Fungsi aktivasi yang digunakan pada layer menjadi faktor penting dalam penentuan keberhasilan akurasi prediksi:

- a. Untuk klasifikasi multi-kelas, seperti 4 kelas jenis lebah, fungsi aktivasi yang digunakan umumnya yaitu softmax. Softmax akan mengubah output dari jaringan menjadi sebuah distribusi probabilitas, di mana setiap nilai output merepresentasikan probabilitas keanggotaan data pada kelas tertentu.
- b. Untuk klasifikasi biner seperti prediksi benar atau salah (0 atau 1), fungsi aktivasi sigmoid umum digunakan dimana fungsi ini akan menghasilkan nilai probabilitas tunggal antara 0 dan 1. Layer inilah yang memberikan hasil akhir yang dapat diinterpretasikan oleh pengguna, seperti label kelas prediksi dari sebuah citra [15].

### 2.1.5.1 MobileNetV2

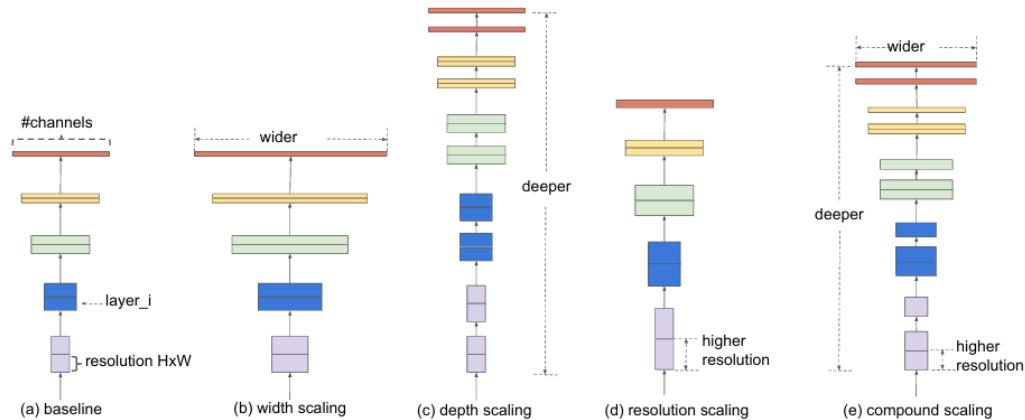


Gambar 2. 6. Arsitektur Base Model MobileNetV2

MobileNetV2 adalah salah satu arsitektur model *pretrained Convolutional Neural Network* (CNN) yang dibangun pada tahun 2018. Model ini dirancang untuk efisiensi tinggi pada perangkat dengan sumber daya komputasi terbatas, seperti *smartphone* [20]. Prinsip arsitektur ini yaitu penggunaan *depthwise separable convolutions*, yang digunakan untuk pemecahan operasi konvolusi menjadi dua langkah terpisah yaitu *depthwise convolution* untuk pemrosesan spasial per kanal dan *pointwise convolution* 1x1 untuk menggabungkan fitur. Mekanisme ini secara signifikan mengurangi jumlah parameter dan beban komputasi. Untuk meningkatkan performa model, MobileNetV2 memiliki dua inovasi utama yaitu *blok inverted residuals* dan *linear bottlenecks* yang bertujuan menjaga akurasi tinggi dan mempertahankan model yang ringan dan cepat [20].

Inovasi MobileNetV2 ini terletak pada blok bangunannya. Struktur *inverted residual* bekerja dengan cara yang berlawanan dari *blok residual konvensional* ia mengambil input berdimensi rendah atau sempit, melebarkannya untuk pemrosesan fitur, lalu mengembalikannya ke dimensi rendah [20]. Sementara itu, *linear bottlenecks* adalah lapisan proyeksi di akhir blok yang tidak menggunakan fungsi aktivasi non-linear seperti ReLU untuk mencegah hilangnya informasi penting pada representasi fitur yang terkompresi. Efisiensi dari model *pretrained* MobileNetV2 mendukung model ini banyak diterapkan dalam berbagai studi kasus di dunia nyata, mulai dari klasifikasi citra hingga analisis citra pertanian yang membantu proses pengolahan.

### 2.1.5.2 EfficientNetB0



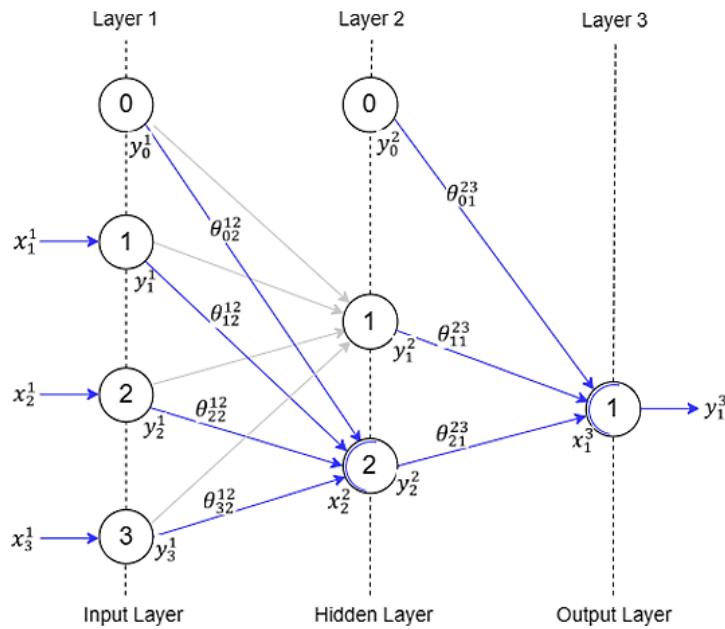
Gambar 2. 7. Arsitektur Base Model EfficientNetB0

EfficientNet adalah salah satu *pretrained* model CNN yang dikembangkan oleh Google pada tahun 2019. Hingga saat ini model EfficientNet memiliki total 8 model yaitu EfficientNetB0 hingga EfficientNetB7 yang memiliki jumlah parameter berbeda dari 5,3 juta hingga 66 juta parameter. EfficientNet dirancang untuk mencapai efisiensi dan akurasi yang tinggi, serta mengatasi metode penskalaan (*scaling*) konvensional yang tidak optimal [21]. Sebelumnya, penskalaan model hanya fokus pada satu dari tiga dimensi, yaitu kedalaman (*depth*), lebar (*width*), atau resolusi gambar, yang terbukti memberikan peningkatan performa yang semakin menurun [21]. Sehingga pada model EfficientNet memperkenalkan metode *compound scaling*, yang secara sistematis menyeimbangkan dan menyeragamkan penskalaan pada ketiga dimensi tersebut secara bersamaan. Metode ini menggunakan satu set koefisien tetap untuk memastikan distribusi sumber daya komputasi yang proporsional ke seluruh bagian jaringan [21]. Arsitektur B0, yang menggunakan blok *mobile inverted bottleneck* (MBConv) sebagai komponen utamanya, kemudian dijadikan cetak biru untuk menghasilkan model-model yang lebih besar dari B1 hingga B7 melalui metode *compound scaling*. Proses ini memungkinkan EfficientNet untuk mencapai akurasi yang lebih unggul dengan jumlah parameter dan kebutuhan FLOPS yang jauh lebih sedikit dibandingkan model-model sebelumnya.

### 2.1.6 Backpropagation

*Backpropagation* merupakan algoritma pelatihan pada jaringan saraf yang mengubah nilai bobotnya menggunakan error output. Terdapat tiga fase dalam pelatihan *backpropagation neural network*, yaitu fase maju (*feed forward*), fase mundur (*backward*), dan fase modifikasi bobot. Dalam fase *feed forward*, pola masukan dihitung dimulai dari lapisan input hingga lapisan output. Dalam fase *backward*, tiap-tiap unit output menerima target pola yang berhubungan dengan pola input untuk dihitung nilai kesalahan (*error*). Nilai *error* ini selanjutnya akan dipropagasi mundur. Sedangkan fase modifikasi bobot bertujuan untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut diulang secara terus menerus hingga kondisi penghentian atau target terpenuhi. Proses *feedforward* dan *backward* dalam *backpropagation* melibatkan penggunaan fungsi *non linearitas* atau fungsi aktivasi. Fungsi aktivasi dapat berupa Sigmoid, ReLU, Tanh, dan lainnya [22]. Berikut penjelasan mengenai *feedforward-pass* dan *backward-pass* beserta komputasi matematisnya:

#### 1. Feedforward Pass



(Sumber: Dzulkarnain, 2024)

Gambar 2. 8. *Feedforward-Pass* Algoritma Backpropagation

Pada Gambar 2.8. proses *feed-forward* merupakan tahapan di mana pola informasi data bergerak maju dari input ke output. Awalnya, nilai input ( $x_{11}$ ,  $x_{21}$ ,  $x_{31}$ ) masuk ke input layer. Kemudian, setiap nilai input akan dikalikan dengan bobotnya ( $\theta$ ) masing-masing dan dijumlahkan di setiap neuron pada hidden layer, lalu ditambah dengan sebuah bobot bias. Hasil penjumlahan lalu diolah oleh fungsi aktivasi untuk menghasilkan output dari hidden layer. Proses yang sama berlanjut, di mana output dari hidden layer dikalikan dengan bobotnya, dan dijumlahkan dengan bobot bias di output layer. Kemudian hasil dilewatkan melalui fungsi aktivasi untuk menghasilkan output akhir. Setelah output didapat, proses *backpropagation* akan menghitung *error* dengan selisih antara output prediksi dengan output target dan akan diproses mundur untuk memperbarui nilai-nilai bobot ( $\theta$ ) agar prediksi selanjutnya menjadi lebih akurat.

Berikut adalah perhitungan matematisnya:

- Input:  $x_1^1 = 1$ ,  $x_2^1 = 1$ ,  $x_3^1 = 1$
- Fungsi Sigmoid  $f(x)$ :  $\frac{1}{1+e^{-x}}$

Example Bobot:

- $\theta_{11}^{12} = 5, \theta_{21}^{12} = 5, \theta_{31}^{12} = 0$ , (bias)  $\theta_{01}^{12} = 0$
- $\theta_{12}^{12} = 6, \theta_{22}^{12} = 4, \theta_{32}^{12} = 0$ , (bias)  $\theta_{02}^{12} = 0$
- $\theta_{11}^{23} = 0.4, \theta_{21}^{23} = -0.5$ , (bias)  $\theta_{01}^{23} = 0.1$

Perhitungan Hidden Layer ( $y_1^2$  dan  $y_2^2$ ):

- Neuron 1

$$\begin{aligned} x_1^2 &= (x_1^1 \cdot \theta_{11}^{12}) + (x_2^1 \cdot \theta_{21}^{12}) + (x_3^1 \cdot \theta_{31}^{12}) + (1 \cdot \theta_{01}^{12}) \\ x_1^2 &= (1 \cdot 5) + (1 \cdot 5) + (1 \cdot 0) + (1 \cdot 0) = 10 \\ y_1^2 &= f(10) = \frac{1}{1+e^{-10}} = 0.99995 \text{ atau } 1 \end{aligned}$$

- Neuron 2

$$\begin{aligned} x_2^2 &= (x_1^1 \cdot \theta_{12}^{12}) + (x_2^1 \cdot \theta_{22}^{12}) + (x_3^1 \cdot \theta_{32}^{12}) + (1 \cdot \theta_{02}^{12}) \\ x_2^2 &= (1 \cdot 6) + (1 \cdot 4) + (1 \cdot 0) + (1 \cdot 0) = 10 \\ y_2^2 &= f(10) = \frac{1}{1+e^{-10}} = 0.99995 \text{ atau } 1 \end{aligned}$$

Perhitungan Output Neuron:

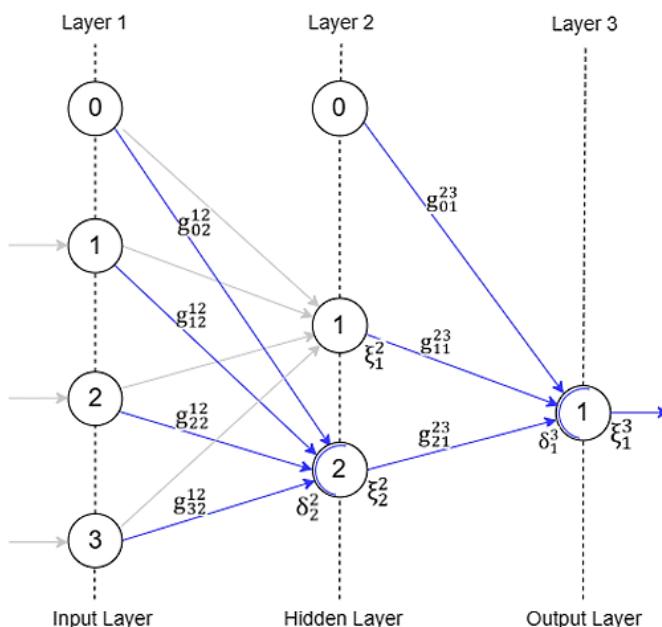
$$x_1^3 = (y_1^2 \cdot \theta_{11}^{23}) + (y_2^2 \cdot \theta_{21}^{23}) + (1 \cdot \theta_{01}^{23})$$

$$x_1^3 = (0.99995 \cdot 0.4) + (0.99995 \cdot -0.5) + (1 \cdot 0.1)$$

$$x_1^3 = 0.39998 - 0.499975 + 0.1 = -0.000005$$

$$y_1^3 = (f(-0.000005)) = \frac{1}{1+e^{-0.000005}} = 0.49998 \text{ atau } 0.5$$

## 2. Backward Pass



(Sumber: Dzurkarnain, 2024 )

Gambar 2. 9. Backward-Pass Algoritma Backpropagation

Pada Gambar 2.9 tahap *backward pass* ini proses ini dimulai dengan menghitung error ( $\delta$ ) pada lapisan output, yaitu selisih antara hasil prediksi dengan target sebenarnya. Kemudian, error ini disebarluaskan mundur (propagasi) ke lapisan sebelumnya untuk menentukan kontribusi error setiap neuron di hidden layer terhadap error akhir. Berdasarkan nilai error di setiap neuron ini, semua bobot ( $g$ ) pada jaringan akan diperbarui bertahap. Proses ini dipengaruhi oleh nilai learning rate dimana agar pada proses feed-forward selanjutnya prediksi output yang dihasilkan lebih akurat.

### 2.1.7 *Supervised Learning*

*Supervised learning* adalah pembelajaran terarah atau terawasi yang mana proses pembelajaran mesin atau komputer dilakukan dengan mempelajari data *training* yang sudah diberi label. *Supervised learning* termasuk ke dalam salah satu cara pembelajaran pada *machine learning* yang dijelaskan sebelumnya. Pembelajaran terarah yang dilakukan ini menggunakan algoritma tertentu dengan pembuatan model berdasarkan data *training* yang digunakan. Label data *training* akan berisi solusi atau output yang diinginkan dari sistem pemodelan yang dibangun. Model pembelajaran terarah (*supervised learning*) biasa digunakan dalam kasus klasifikasi dan regresi dalam *machine learning*. Adapun Algoritma yang tergolong ke dalam *supervised learning* contohnya seperti *Regresi Linier*, *Regresi Logistic*, *Linear Discriminant Analysis*, *K-Nearest Neighbors*, *Support Vector Machines*, *Random Forest*, dan *Neural Network* [23].

Tujuan utama dalam pembelajaran *supervised* ini adalah menemukan sebuah fungsi atau model yang dapat mendekati fungsi dasar atau output sebenarnya berdasarkan dataset yang dilatih. Tahapan utama dalam *supervised learning* dibagi menjadi *representative*, optimasi, dan generalisasi. Langkah pertama adalah memilih keluarga model yang cukup kompleks seperti *neural network* untuk merepresentasikan hasil atau output yang diinginkan. Selanjutnya, parameter model diidentifikasi dengan meminimalkan fungsi *loss* tertentu. Kemudian, fungsi terbaik yang ditemukan akan digunakan untuk memprediksi dataset uji, jika terdapat kesalahan prediksi maka disebut sebagai *test error*. Kesalahan ini dapat dilihat dari tiga aspek seperti *representasi error* yang terjadi karena keterbatasan model dalam merepresentasikan fungsi, *optimization error* yang terjadi akibat kegagalan optimasi dalam menemukan parameter optimal, dan *generalization error* yang terjadi karena ketidakmampuan model beradaptasi ke data baru atau data uji [24].

### 2.1.8 *Citra Digital*

Citra digital adalah data matriks dua dimensi memuat indeks baris dan kolom yang mengandung informasi intensitas warna dengan format RGB (*Red*, *Green*, *Blue*) yang setiap piksel memiliki komponen intensitas  $I_R$ ,  $I_G$ ,  $I_B$  [25]. Suatu

citra warna dapat direpresentasikan oleh matriks  $I_P$  yang merepresentasikan nilai derajat keabuan pada masing-masing *channel red, green, dan blue*.  $I_R$  merupakan matriks yang merepresentasikan derajat keabuan citra pada *channel Red*,  $I_G$  merupakan matriks yang merepresentasikan derajat keabuan citra pada *channel Green*, dan  $I_B$  merupakan matriks yang merepresentasikan derajat keabuan citra pada *channel Blue* [26].

$$I_P = \begin{bmatrix} (a_{11})_p & (a_{12})_p & \cdots & (a_{1C})_p \\ (a_{21})_p & (a_{22})_p & \cdots & (a_{2C})_p \\ \vdots & \vdots & \ddots & \vdots \\ (a_{B1})_p & (a_{B2})_p & \cdots & (a_{BC})_p \end{bmatrix}$$

Nilai untuk setiap komponen  $I_P$  adalah sebagai berikut:

$$(a_{ij})_p = \frac{d_p}{255}$$

Keterangan:

- B : jumlah piksel baris pada citra
- C : jumlah piksel kolom pada citra
- $d_p$  : nilai diskrit representasi derajat keabuan citra pada *channel* ke  $p$  yang memiliki range antara 0 hingga 255

Citra digital memiliki keunggulan dari data lain, karena kemampuannya untuk mendekripsi dan mengenali objek dari jarak jauh dengan area yang luas (*wide view*) tanpa perlu menyentuh (*non-contact*) ataupun merusaknya (*non-destructive*). Dalam keunggulannya citra digital dapat diimplementasikan dalam berbagai bidang seperti dalam analisis produk, medis, maupun pertanian. Dalam penerapannya dilakukan serangkaian langkah yang disebut pengolahan citra digital. Adapun langkah dalam pengolahan citra digital terbagi dalam 3 langkah yaitu konversi warna, segmentasi, dan filter [25]. Istilah pengolahan citra digital sendiri berarti memproses atau mengolah gambar dua dimensi menggunakan komputer. Pengolahan citra digital dapat berupa gradasi, modifikasi, maupun manipulasi citra untuk tujuan tertentu. Pengolahan citra menjadi bagian penting yang mendasari berbagai aplikasi nyata, seperti pengenalan pola, penginderaan jarak-jauh melalui satelit atau pesawat udara, dan *machine vision* [27].

### 2.1.9 Dataset

Dataset adalah kumpulan banyak data dengan jumlah yang bervariasi. Dataset dalam *deep learning* digunakan untuk melatih, menguji, dan mengevaluasi model. Dataset ini biasanya terdiri dari berbagai fitur variabel input dan label variabel output yang relevan dengan masalah yang ingin diselesaikan. Dalam implementasinya, kualitas dan keberagaman dataset harus diperhatikan karena model *deep learning* sangat bergantung pada dataset yang diberikan untuk pelatihan pada model agar model dapat belajar dengan lebih baik. Dataset yang baik harus mencakup data yang *representatif*, bebas dari noise, dan memiliki distribusi yang seimbang untuk menghindari bias agar model dapat belajar pola yang tepat. Dataset biasanya melalui beberapa tahapan sebelum dilatih pada model yaitu meliputi proses pengumpulan, pembersihan, dan *pra-pemrosesan* data seperti normalisasi, *labeling*, dan augmentasi untuk memastikan dataset siap. Sebelum pelatihan model, dataset biasanya dibagi menjadi *subset* data pelatihan (*training*), data validasi (*validation*), dan data uji (*testing*) untuk mengevaluasi kinerja model secara objektif sehingga cukup optimal untuk digunakan [17].

### 2.1.10 Library dan Framework

*Library* atau perpustakaan adalah kumpulan kode program yang dapat langsung digunakan berisi fungsi, modul, atau algoritma untuk memudahkan pengembangan model. *Library* biasanya sudah tersedia dalam bahasa pemrograman, namun ada juga *library* tambahan yang perlu diinstal pengguna terlebih dahulu. *Library* dapat digunakan oleh pengguna dengan memanggil fungsi atau modul yang sudah ada sebelumnya. Adapun *library* yang umum digunakan dalam bahasa pemrograman Python untuk *deep learning* seperti TensorFlow, PyTorch, dan Scikit-Learn [28]. Selain itu ada juga *library* seperti NumPy dan Pandas yang digunakan dalam pengolahan dan manipulasi data matematika atau numerik. Keunggulan *library* sendiri terletak pada modularitasnya, dimana kemudahan dimana pengguna hanya perlu memanggil fungsi yang diperlukan tanpa mengatur struktur keseluruhan proyek.

*Framework* atau kerangka kerja adalah struktur dasar yang berisi banyak kode program atau *library* dan *framework* biasanya sudah menentukan aturan, dan

workflow yang harus diikuti dalam pengembangan sistem atau model, sehingga tidak harus memulai dari nol. Salah satu contoh *framework* yang sering digunakan dalam *deep learning* seperti Keras yang terintegrasi dengan *library* TensorFlow dengan menyediakan lapisan abstraksi tinggi dalam membangun jaringan saraf dengan cepat, sekaligus mengatur aliran dan susunan komponen data [13]. *Framework* yang sudah diinstall dapat dijalankan dengan mengikuti template dari struktur direktori yang sudah ada atau dengan meng-import model yang sudah disediakan. Seperti pada *framework* PyTorch *Lightning* dengan standar proses pelatihan model dengan mengotomatiskan distribusi GPU dan *logging*, sehingga mengurangi duplikasi kode [28].

### **2.1.11 Teknik Augmentasi**

Augmentasi adalah teknik yang digunakan untuk meningkatkan variasi dataset dengan menciptakan variasi data dinamis dengan mempertahankan label aslinya. Teknik augmentasi ini dapat digunakan untuk meningkatkan akurasi model dan mencegah terjadinya *overfitting* ataupun *underfitting*. Disamping itu, penerapan teknik augmentasi ini perlu dilakukan dengan tepat untuk menghindari resiko model belajar dari data yang tidak representatif bahkan salah [29]. Dalam beberapa penelitian telah disebutkan bahwa teknik augmentasi data untuk memperkaya dataset ini berpengaruh terhadap peningkatan prediksi hasil akurasi model yang dilatih. Penelitian sebelumnya yang membagi augmentasi menjadi metode dasar seperti *rotation*, *cropping*, dan *flipping* serta metode lanjutan seperti *auto augment*, augmentasi fitur, dan generatif model seperti GAN menerangkan bahwa penelitian menunjukkan hasil peningkatan akurasi [30]. Sementara pada penerapan teknik augmentasi citra lainnya yaitu *image smoothing*, *image sharpening*, *contrast stretching*, dan *image brightening* juga meningkatkan akurasi model dari 54,9% menjadi 99,6% dari 2.801 test data [31].

### **2.1.12 Validasi dan Optimalisasi**

*Validation* (validasi) model adalah tahap yang dilakukan selama proses pelatihan untuk memastikan atau memvalidasi kinerja model yang dilatih. Selama proses pelatihan atau *training*, model akan divalidasi kinerjanya pada data baru,

validasi model biasanya bertujuan untuk menghindari *overfitting* (model terlalu menghafal data latih) dan *underfitting* (generalisasi model kurang baik). Dalam validasi model selama pelatihan biasanya menggunakan metode *cross validation* (CV) ataupun *holdout*. Pada metode CV model yang dilatih dengan subset data training akan divalidasi dengan subset data validation dan akan divalidasi dalam beberapa kali dengan kombinasi data yang berbeda. Jenis *CV* pun juga beragam disesuaikan dengan ukuran dataset seperti *k-fold cross-validation*, *stratified cross-validation*, dan lainnya. Sementara teknik *Holdout* akan membagi dataset menjadi subset data *training* dan subset data uji, lalu model hanya diuji sekali [23].

*Optimizing* (optimalisasi) model adalah teknik yang diterapkan untuk meningkatkan performa model dengan menyesuaikan parameter, arsitektur, epoch atau proses pelatihan. Optimalisasi model dapat dilakukan selama proses training atau setelah evaluasi yang biasa dilakukan untuk meningkatkan akurasi, meminimalkan kesalahan, dan meningkatkan efisiensi pelatihan. Optimalisasi ini meliputi berberapa teknik seperti penggunaan optimizer seperti Adam, *SGD (Stochastic Gradient Descent)*, *RMSProp* yang sudah umum untuk peningkatan akurasi model [30]. Teknik regularisasi seperti *dropout* atau *batch normalization*, dan penyesuaian *hyperparameter* seperti memilih *learning rate*, ukuran *batch*, atau variasi bobot berdasarkan jumlah lapisan (*layers*) yang tepat juga penting untuk mencegah masalah *gradient explosion* atau *vanishing* [24].

### 2.1.13 Metrik Evaluasi

Metrik evaluasi adalah suatu metode yang digunakan untuk mengevaluasi model atau mengukur performa model yang dihasilkan dari proses training atau pelatihan. Metrik evaluasi ini memiliki beberapa jenis yang disesuaikan dengan jenis *task* atau kasus yang dikerjakan dan informasi apa yang ingin diketahui. Pada umumnya terdapat empat metrik evaluasi yang umum digunakan yaitu *Confusion Matrix*, *ROC-AUC*, *Root Mean Square Error (RSME)*, dan *R-Square*. Dalam tugas klasifikasi, metrik yang sering digunakan yaitu *Confusion Matrix* dan *ROC-AUC*. *Confusion matrix* adalah metrik evaluasi berukuran N x N, dimana N adalah jumlah kelas yang diprediksi. *Confusion matrix* akan membandingkan antara hasil prediksi dan hasil yang sebenarnya, lalu akan ditampilkan ringkasan semua hasil prediksi.

Sementara ROC menampilkan grafik probabilitas dan AUC menampilkan derajat keterpisahan. Berikut salah satu gambaran metrik  $2 \times 2$  dengan kelas *his* dan *her* pada *confusion matrix* [23].

Tabel 2. 2. Contoh Tabel *Confusion Matrix*

		Actual = His (1)	Actual = Her (0)
Predicted = His (1)	TP	FP	
	FN	TN	
Predicted = Her (0)			

Dari nilai *Confusion matrix* di atas dapat dihitung nilai metrik evaluasi berikut:

1. *True Positif Rate (Recall)* =  $\frac{TP}{TP + FN}$
2. *True Negative Rate (Specificity)* =  $\frac{TN}{TN + FP}$
3. *False Positif Rate (FPR atau Type I Error)* =  $\frac{FP}{FP + TN} = 1 - Specificity$
4. *False Negative Rate (FNR atau Type II Error)* =  $\frac{FN}{FN + TP} = 1 - Recall$
5. *Precision* =  $\frac{TP}{TP + FP}$
6. *F1-Score* =  $2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$
7. *Accurature* =  $\frac{TP+TN}{TP+TN+FP+FN}$

*Confusion matrix* akan menunjukkan hasil prediksi model ke dalam empat kategori yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)* yang dapat menunjukkan distribusi hasil prediksi model benar dan salah. Adapun rincian setiap kondisi tabel 2.2 yaitu sebagai berikut:

1. TP (*True Positive*) berisi jumlah data points diberi label ‘His’ yang memang sebenarnya bernilai ‘His’.
2. TN (*True Negative*) berisi jumlah data points diberi label ‘Her’ yang memang sebenarnya bernilai ‘Her’.
3. FP (*False Positive*) berisi jumlah data points diberi label ‘His’ yang sebenarnya bernilai ‘Her’. Error ini biasa disebut *Error Type 1*.

4. FN (*False Negative*) berisi jumlah data points diberi label ‘Her’ yang sebenarnya bernilai ‘His’. Error ini biasa disebut *Error Type 2*.
5. *Accurature* adalah perhitungan performa model yang membandingkan jumlah data yang diprediksi benar dengan total seluruh prediksi yang dilakukan.
6. *Precision* adalah perhitungan untuk membandingkan jumlah data yang diprediksi sebagai positif secara benar terhadap jumlah data yang diprediksi positif.
7. TPR (*Recall*) adalah perhitungan untuk membandingkan jumlah data yang diidentifikasi benar terhadap seluruh data yang sebenarnya memang benar.
8. TNR (*Specificity*) adalah perhitungan untuk membandingkan jumlah data yang diidentifikasi negatif secara benar terhadap data yang sebenarnya memang salah.
9. FPR (*Type Error I*) adalah perhitungan untuk membandingkan jumlah data yang salah identifikasi sebagai positif dengan jumlah data yang dibenar-benar negatif.
10. FNR (*Type Error II*) adalah perhitungan untuk membandingkan jumlah data yang salah identifikasi sebagai negatif dengan jumlah data yang dibenar-benar positif.
11. F1-Score adalah pehitungan yang digunakan untuk menghitung rata-rata harmonik (*harmonic average*) dari *precision* dan *recall*.

Hasil tabel *confussion matrix* ini diolah dengan *library scikit learn* untuk mendapatkan perhitungan akhir *classification report* dan *accuracy*. Perhitungan nilai dilakukan oleh pemanggilan fungsi-fungsi pada *scikit learn* untuk *classification report*, dan *accuracy score*. Hasil yang didapatkan kemudian akan ditampilkan dengan penggunaan fungsi pemanggilan .plt dari dari *library matplotlib* dan *seaborn* untuk membuat grafik seperti *heatmap*.

## 2.2 Penelitian Terdahulu

Dalam melaksanakan penelitian ini, dilakukan analisis dari beberapa studi literatur terkait dari penelitian terdahulu yang relevan dengan topik pengembangan model klasifikasi dengan arsitektur *Convolutional Neural Network (CNN)* berdasarkan dataset citra yang digunakan. Analisis yang dilakukan menjadi kajian literatur dan referensi dalam pengembangan model ini. Berikut merupakan beberapa penelitian terdahulu yang relevan dengan topik penelitian ini.

Penelitian yang dilaksanakan oleh Showmick Guha Paul *et al.*, pada tahun 2023 yang berjudul *A real-time application-based convolutional neural network approach for tomato leaf disease classification*. Penelitian ini mengembangkan model diagnosis klasifikasi jenis penyakit daun tomat yang terdiri atas 11 kelas termasuk 1 kelas daun sehat. Metode yang digunakan dalam penelitian ini adalah lightweight custom *Convolutional Neural Network (CNN)* model dan *utilized Transfer Learning* dengan arsitektur model *VGG-16* and *VGG-19*. Optimizer yang digunakan adalah Adam. Hasil dari penelitian ini menunjukkan akurasi model terbaik mencapai 95% diperoleh dari model kustom dengan penyesuaian arsitektur 4 lapisan konvolusi, 3 *dropout*, 1 *fully connected layer* mencapai setelah augmentasi data, mengungguli *VGG-16* pada akurasi 92% dan *VGG-19* dengan akurasi 89%. Model ini dapat di *deployment* secara *end-to-end* (E2E) berbasis website dan berbasis android atau mobile untuk klasifikasi *real-time* [6].

Penelitian yang dilaksanakan oleh Jose Miguel Valiente *et al.*, pada tahun 2023 yang berjudul *Automatic pollen recognition using convolutional neural networks: The case of the main pollens present in Spanish citrus and rosemary honey*. Penelitian ini bertujuan mengidentifikasi dan klasifikasi serbuk sari. Metode yang digunakan dalam penelitian ini adalah lightweight custom *Convolutional Neural Network (CNN)* dengan dataset terdiri dari 18.864 gambar mikroskopik 16 jenis serbuk sari/partikel. Penelitian ini membandingkan akurasi pengujian 8 arsitektur CNN *pre-trained* yaitu VGG16, VGG19, InceptionV3, Xception, ResNet50, DenseNet201, MobileNetV2, dan EfficientNetV2M serta model baru PolleNetV1 dengan arsitektur sederhana 6 lapisan konvolusi. Hasil dari penelitian ini menunjukkan akurasi model terbaik dicapai oleh InceptionV3 dengan akurasi 97,99%, sementara PolleNetV1 mencapai akurasi 95,66% dengan penggunaan optimizer SGD serta teknik *augmentasi data* [32]

Penelitian yang dilaksanakan oleh William Kelley *et al.*, pada tahun 2021 yang berjudul *Honey sources: neural network approach to bee species classification*. Penelitian ini bertujuan mengembangkan model klasifikasi otomatis tiga spesies lebah yaitu *Apis mellifera*, *Bombus griseocollis*, *Bombus impatiens* dari dataset crowdsource BeeSpotter sejumlah 15.347 gambar dari *crowdsourcing*. Metode yang digunakan adalah kombinasi *Faster R-CNN* dengan backbone *Resnet 101+FPN* untuk deteksi objek lebah dan klasifikasi spesies. Penelitian memanfaatkan *optimizer SGD* dan teknik augmentasi data seperti *random flip* dan *crop*. Hasil dari penelitian ini menunjukkan akurasi klasifikasi mencapai 91% untuk tiga spesies lebah yang dilatih pada dataset [33]

Penelitian yang dilaksanakan oleh Keanu Buschbacher *et al.*, pada tahun 2019 yang berjudul *Image-based species identification of wild bees using convolutional neural networks*. Penelitian ini mengembangkan *DeepABIS*, sistem identifikasi spesies lebah dan kupu-kupu berbasis CNN yang ringan untuk aplikasi lapangan. Penelitian menggunakan arsitektur MobileNetV2 dengan transfer learning, model ini diimplementasikan dalam aplikasi bebas mobile yaitu *MobileABIS* dan web pada *CloudABIS*. Dataset utama terdiri dari 7.595 citra sayap lebah dari 124 spesies dan 300 citra kupu-kupu. Pelatihan model menggunakan augmentasi data dan *class weighting* untuk menangani ketidakseimbangan kelas. Hasil menunjukkan akurasi top-1 93,95% untuk data lebah dan 96,72% untuk data kupu-kupu, dengan waktu inferensi 301 ms di perangkat mobile. Keunggulan utama meliputi efisiensi komputasi, adaptabilitas, serta integrasi platform partisipatif berbasis cloud [34].

Penelitian yang dilaksanakan oleh Domingos Alves Dias Júnior *et al.*, pada tahun 2021 yang berjudul *Automatic method for classifying COVID-19 patients based on chest X-ray images, using deep features and PSO-optimized XGBoost*. Penelitian ini bertujuan mengembangkan sistem otomatis untuk mengklasifikasikan pasien COVID-19 melalui citra *X-ray* dada dengan menggabungkan ekstraksi *deep features* dan klasifikasi *XGBoost* yang dioptimalkan menggunakan algoritma *Particle Swarm Optimization* (PSO). Dataset terdiri dari 206 citra COVID-19 dan 1.341 citra normal yang dilakukan *preprocessing* seperti konversi grayscale dan resizing 224x224. Fitur diekstraksi menggunakan perbandingan

arsitektur VGG19, Inception-v3, dan ResNet50, lalu diklasifikasi dengan *XGBoost* yang hyperparameternya dioptimalkan dengan PSO. Hasil terbaik dicapai oleh VGG19 + XGBoost + PSO dengan akurasi 98,71%, presisi 98,89%, *recall* 99,63%, dan F1-score 99,25%. Penelitian ini menunjukkan PSO terbukti lebih efektif dibandingkan *Genetic Algorithm* (GA) dan *Bee Colony Optimization* (BCO) dalam meningkatkan kinerja model. Penelitian ini juga menunjukkan keunggulan *deep features* dibandingkan metode tradisional seperti *texture features* [35].

Penelitian yang dilaksanakan oleh Dicki Irfansyah dkk., pada tahun 2021 yang berjudul *Arsitektur Convolutional Neural Network (CNN) Alexnet untuk Klasifikasi Hama pada Citra Daun Tanaman Kopi*. Penelitian ini berfokus pada pengembangan model identifikasi klasifikasi penyakit daun tanaman kopi. Metode yang digunakan yaitu arsitektur CNN AlexNet. Dataset ini mencakup total 300 data yang terdiri dari tiga kelas yaitu *health*, *rust*, dan *red spider mite* dengan pembagian 260 data untuk pelatihan dan 40 data untuk pengujian. Hasil penelitian menunjukkan bahwa akurasi pelatihan atau *training* data berada di rentang 69,44% - 80,56% dan akurasi pengujian mencapai 81,6% [36].

Penelitian yang dilaksanakan oleh Waeisul Bismi dkk., pada tahun 2023 yang berjudul *Klasifikasi Citra Genus Panthera Menggunakan Pendekatan Deep Learning Berbasis Convolutional Neural Network (CNN)*. Penelitian ini bertujuan mengembangkan model klasifikasi citra genus *Panthera* menggunakan metode *Convolutional Neural Network (CNN)*. Dataset penelitian ini terdiri dari 6.290 citra yang dibagi menjadi 5.032 data latih, 629 validasi, dan 629 uji data ini mencakup enam kelas dari genus *Panthera* yaitu harimau, singa, macan tutul, jaguar, liones, dan snow tiger. Pembuatan model CNN juga melibatkan proses pra-pemrosesan citra seperti *resizing*, konversi warna, dan normalisasi. Hasil penelitian dengan penggunaan *optimizer* Adam menunjukkan bahwa model *Convolutional Neural Network (CNN)* mencapai akurasi klasifikasi sebesar 85,21% dalam membedakan citra genus *Panthera* [37].

Penelitian yang dilaksanakan oleh Isna Wulandari dkk., pada tahun 2020 yang berjudul *Klasifikasi Citra Digital Bumbu dan Rempah dengan Algoritma Convolutional Neural Network (CNN)*. Penelitian ini bertujuan untuk mengembangkan model klasifikasi citra untuk mengidentifikasi tiga jenis rempah

yaitu ginseng, jahe, dan lengkuas menggunakan algoritma *Convolutional Neural Network (CNN)*. Dataset penelitian terdiri dari 300 citra digital yang diperoleh melalui *web crawling* dengan proporsi perbandingan data pelatihan dan pengujian sebesar 80:20. Model *Convolutional Neural Network (CNN)* yang digunakan memiliki 2 lapisan konvolusi dengan masing-masing 10 filter pada lapisan pertama dan 20 filter pada lapisan kedua. Hasil penelitian menunjukkan akurasi data pelatihan sebesar 98,75% dengan nilai *loss* 0,0769, sedangkan akurasi data pengujian mencapai 85% dengan nilai *loss* 0,4773. Pengujian data uji baru menggunakan 3 citra tiap kelas menghasilkan akurasi sebesar 88,89% [26].

Penelitian yang dilaksanakan oleh Achmadi dkk., pada tahun 2024 yang berjudul *Identifikasi Penyakit Daun Tembakau Berbasis Pengolahan Citra dengan Metode Convolutional Neural Network (CNN) dan Metode Transfer Learning*. Penelitian ini bertujuan untuk mengidentifikasi penyakit pada daun tembakau. Penelitian ini menggunakan metode arsitektur CNN *pre-trained* MobileNet dengan pendekatan transfer learning. Dataset yang digunakan terdiri dari 261 citra daun tembakau yang terbagi menjadi tiga kelas yaitu daun sehat, busuk daun, dan mosaik daun. Data ini dibagi dengan proporsi perbandingan data latih 80%, uji 10%, dan validasi 10%. Hasil penelitian menunjukkan bahwa model mencapai nilai precision sebesar 73%, recall 69%, F1-score 68%, dan akurasi keseluruhan sebesar 69% pada data uji [38].

Penelitian yang dilaksanakan oleh Ar'rafi Akram dkk., pada tahun 2023 yang berjudul *Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode Convolutional Neural Network*. Penelitian ini bertujuan mengembangkan model klasifikasi otomatis untuk 13 jenis hama serangga dalam sektor pertanian. Metode yang digunakan dalam penelitian ini adalah *Convolutional Neural Network (CNN)* dengan arsitektur Xception. Dataset yang digunakan terdiri dari 1.363 citra yang dibagi menjadi citra data latih 1090, uji 136, dan validasi 137. Hasil penelitian dengan *optimizer* Adam menunjukkan bahwa model mencapai akurasi sebesar 93,81% pada data pelatihan, 81,75% pada data validasi, dan akurasi pada data uji 80% dengan nilai *precision* 73%, *recall* 69%, dan F1-score 68% [39].

Beberapa penelitian yang telah dijabarkan sebelumnya telah membantu peneliti dalam proses pelaksanaan penelitian ini. Literatur terkait metode

pengembangan, dataset, tahap *processing* data, tahap evaluasi model, tahap validasi model, tahap pengujian model, dan hasil uji menjadi acuan dan referensi penting bagi peneliti untuk mengembangkan model klasifikasi lebah madu dalam penelitian ini sesuai kebutuhan dan keterbatasan penelitian yang ada.

### 2.3 Analisis Penelitian Serupa

Berdasarkan penelitian yang dilakukan oleh Syifa Fhadhillah Chairunissa pada tahun 2023 yang berjudul “Klasifikasi Jenis Lebah Madu Menggunakan Metode Convolutional Neural Network” [39]. Penelitian bertujuan untuk mengembangkan sistem klasifikasi jenis lebah menggunakan metode Convolutional Neural Network (CNN). Data yang digunakan dalam penelitian ini berupa dataset citra lebah madu dari jenis Carnolian Honey Bee, Italian Honey Bee, dan Russian Honey Bee. Dalam penelitian dilakukan pre-processing citra seperti resizing dan normalisasi piksel dengan model CNN menggunakan optimizer Adam dan perbandingan dengan optimizer SGD. Hasil evaluasi model yang dilatih menunjukkan bahwa model CNN dengan optimizer Adam memiliki akurasi sebesar 98% dalam mengklasifikasikan citra lebah yang digunakan [40].

Berdasarkan penelitian terkait klasifikasi lebah madu ini, ditemukan beberapa poin bahwa metode Convolutional Neural Network (CNN) memberikan hasil yang cukup akurat dalam kasus klasifikasi citra lebah. Dalam penelitian juga diketahui bahwa jumlah dataset penelitian yang digunakan masih terbatas, selain itu dalam penelitian ini belum diterapkannya teknik augmentasi data yang mana teknik ini dapat digunakan untuk memperluas variasi dataset yang menurut beberapa literatur, teknik ini dapat meningkatkan akurasi pelatihan model terutama pada dataset terbatas. Maka dari itu, dalam penelitian ini mengangkat klasifikasi lebah madu tanpa sengat (*stingless*) yang mana dari literatur disebutkan bahwa potensi yang dimiliki lebah madu *stingless* ini lebih tinggi, selain itu penelitian ini akan menggunakan dataset berupa citra yang diambil langsung dari Lembah Suhita. Dalam penelitian ini akan dilakukan penerapan model *pre-trained* CNN dengan *transfer learning* dan proses *fine-tuning* jika akurasi yang dihasilkan kurang optimal. Hasil pelatihan model terbaik akan diuji atau evaluasi pada dataset uji yang telah dibagi sebelumnya. Kemudian akan digunakan juga teknik augmentasi data

yang akan memperluas dataset citra lebah yang diambil. Sehingga harapan penelitian ini dapat menambah dataset yang ada dalam objek penelitian lebah madu khususnya jenis *stingless honey bee*, dan penelitian ini akan mencapai hasil yang cukup akurat dari hasil proses *training* yang dilakukan.

## 2.4 Tools yang Digunakan

### 2.4.1 Jupyter Notebook

Jupyter Notebook adalah sebuah platform web open-source yang dapat diakses secara online oleh pengguna untuk membuat dokumen yang disebut ‘notebooks’ berisi kode, simbol, visualisasi, atau teks penjelasan. Jupyter Notebook cocok digunakan dalam proses data *cleaning*, pemodelan *statistic*, data visualisasi, *machine learning*, dan lainnya [41]. *Platform* jupyter ini mendukung lebih dari 40 bahasa pemrograman, termasuk Python, R, Julia, dan Scala, sehingga sangat fleksibel untuk analisis data, pembelajaran mesin, dan komputasi ilmiah. Selain itu, Jupyter Notebook memungkinkan pengguna untuk membagikan notebooks untuk kolaborasi tim melalui email, Dropbox, GitHub, ataupun Jupyter Notebook Viewer. Jupyter Notebook juga telah menyediakan beragam output interaktif yang dapat dipilih pengguna baik berupa HTML, images, videos, LaTeX, dan kompatibilitas dengan platform cloud seperti Google Colab dan AWS SageMaker untuk akses jarak jauh dan komputasi berskala besar. Jupyter Notebook menyediakan integrasi dengan alat berbasis big data seperti Apache Spark dan mendukung untuk visualisasi data terintegrasi dengan library seperti Matplotlib, Seaborn, dan Plotly yang memungkinkan dalam pembuatan grafik langsung.

### 2.4.2 Google Colab

Google Colab atau ‘Colaboratory’ adalah salah satu *platform* dari Google *research* yang dapat diakses pengguna untuk menulis dan menjalankan kode program python melalui browser tanpa memerlukan konfigurasi tambahan. Google Colab menyediakan layanan *cloud* gratis melalui Google *hosting* untuk mendukung pembelajaran *machine learning*, data *analysis*, maupun *artificial intelligence* yang sering kali memerlukan kemampuan komputasi yang cukup besar. Google Colab

juga menyediakan akses gratis bagi pengguna untuk menggunakan GPU dan TPU yang sering menjadi masalah dalam menyelesaikan permasalahan komputasi yang besar. Selain akses gratis Google Colab juga menyediakan layanan Pro dan Pro+ yang tentunya akan lebih menguntungkan pengguna dari aspek prioritas GPU dan TPU, RAM, dan kestabilan sesi maupun jaringan. Google Colab telah menyediakan hampir semua modul yang dibutuhkan pengguna dalam data *science* atau *analysis*, seperti modul atau *library* Pandas, Numpy, Scipy, dan selain itu juga terdapat *framework deep learning* seperti Tensorflow, Keras, OpenCV dan Pytorch. Selain itu Google Colab dapat digunakan secara bersama untuk kolaborasi kerja antar tim, kemudian file *notebooks* yang sudah dibuat pada Google Colab dapat *import, publish, ataupun save* ke berbagai *platform* seperti Google Drive ataupun Github [41].

### 2.4.3 Python

Python adalah salah satu bahasa pemrograman dinamis, tingkat tinggi, dan dikenal sebagai bahasa program interpreter yaitu bahasa program yang mengkonversikan source code menjadi machine code secara langsung saat program dijalankan. Pendeklarasian variabel pada bahasa python tidak perlu menggunakan tipe data maupun penggunaan titik koma pada blok kode seperti halnya bahasa program lain karena python bersifat dinamis tetapi mudah dipelajari serta menyediakan banyak struktur data tingkat tinggi untuk berkabagi keperluan pengembangan perangkat lunak termasuk pada pemrograman berorientasi objek hingga komputasi data besar [42]. Python juga memiliki pustaka bawaan yang sangat luas, mencakup berbagai bidang seperti *data science*, *machine learning*, pemrosesan gambar, pengembangan game, dan lainnya. Contohnya, pustaka seperti Pandas dan NumPy digunakan untuk analisis data dan komputasi ilmiah, sementara TensorFlow dan PyTorch populer dalam *deep learning*. Keunggulan Python terletak pada fleksibilitasnya untuk berbagai aplikasi komputasi modern serta kemampuannya untuk mengurangi kompleksitas kode secara signifikan [43].

#### 2.4.4 TensorFlow dan Keras

TensorFlow adalah *framework* sekaligus *library* bersifat *open-source* yang dikembangkan oleh Google Brain Team. TensorFlow ini dapat digunakan dalam membangun atau melatih model dalam *machine learning*. TensorFlow memiliki infrastruktur yang cukup kuat untuk berbagai jenis pemodelan dalam AI. TensorFlow bersifat fleksibel dan dapat digunakan di berbagai platform. Adapun kelebihan dari TensorFlow yaitu dapat digunakan dengan mudah untuk membuat model AI skala kecil hingga besar. Selain itu fleksibilitasnya dalam mendukung berbagai model deep learning dan machine learning. TensorFlow juga memiliki kelebihan dalam fitur optimisasi kinerja untuk deployment di berbagai platform, yang memungkinkan optimisasi kinerja menggunakan teknologi seperti TensorRT. Adapun cara kerja TensorFlow yaitu menggunakan konsep graf komputasi, di mana operasi matematika digambarkan sebagai simpul (*node*) yang terhubung oleh tepian (*edge*). Model AI dibangun dalam bentuk graf komputasi dan TensorFlow membantu mengoptimalkan eksekusi graf dengan membangun graf komputasi yang menggambarkan hubungan antar operasi seperti confussion matriks atau penghitungan error. Saat graf siap, TensorFlow akan menganalisisnya untuk menghapus langkah yang tidak perlu, mengurutkan operasi yang saling bergantung, dan menemukan operasi yang bisa dijalankan secara paralel misalnya, dua perhitungan matematis independen bisa berjalan bersamaan di CPU/GPU berbeda. Teknik ini, yang disebut evaluasi lalai yang memungkinkan TensorFlow menggunakan sumber daya secara efisien sekaligus mempercepat eksekusi. Sehingga, komputasi jadi lebih cepat dan hemat sumber daya tanpa perlu pengaturan manual dari pengguna terutama untuk komputasi skala besar [11].

Keras adalah *high-level API* berbasis Python untuk membangun dan melatih model *deep learning*, dirancang untuk memudahkan prototyping dengan prinsip modularitas dan kemudahan penggunaan. Sejak diintegrasikan sebagai bagian TensorFlow Keras menyediakan antarmuka yang intuitif untuk mengimplementasikan arsitektur kompleks seperti CNN, RNN, atau *transformer* dengan hanya beberapa baris kode, dengan tetap mendukung *customization*. Keras menekankan *user experience* yang konsisten, memungkinkan *user* fokus pada eksperimen model tanpa kompleksitas *low-level* [15]. Selain itu, Keras dikenal

dengan kemampuannya dalam mempercepat pengembangan model pada klasifikasi citra, pemrosesan bahan alami dengan dukungan *tools* seperti *callbacks*, *pre-trained models*, dan integrasi *TensorFlow Ecosystem* [44].

#### 2.4.5 Scikit-Learn

Scikit-learn adalah *library* Python yang populer dalam *machine learning* tradisional yang cocok untuk membangun model seperti klasifikasi, regresi, atau clustering dengan cepat. Keunggulannya terletak pada antarmuka sederhana dan intuitif misalnya, model dapat dilatih hanya dengan beberapa baris kode serta dukungan algoritma klasik seperti SVM, Random Forest, atau PCA. Selain itu Scikit-learn dapat menampilkan antarmuka yang konsisten untuk berbagai algoritma pembelajaran mesin [11]. Adanya integrasi dengan NumPy, Pandas, dan Matplotlib juga memudahkan penyusunan alur kerja lengkap, dari *preprocessing* data hingga evaluasi akurasi. Scikit-learn lebih berfokus pada algoritma machine learning klasik. Sehingga dalam kasus pemodelan *deep learning* yang lebih kompleks membutuhkan *framework* yang lebih khusus dan teroptimasi seperti TensorFlow atau Keras. Dalam hal ini, scikit-learn tidak digunakan sebagai *library* utama dalam penelitian ini, karena fokus implementasi pada *Convolutional Neural Network* (CNN) untuk klasifikasi citra pada jenis lebah. Tetapi, scikit-learn tetap dimanfaatkan dalam beberapa tahap pendukung lainnya seperti *preprocessing* data, evaluasi metrik kinerja, validasi model, dan perbandingan performa model yang memungkinkan pengguna mengimpor model, melatihnya data, dan melakukan prediksi.

#### 2.4.6 GitHub

GitHub adalah platform web berbasis cloud yang populer untuk pengembang program dalam menyimpan, mengelola, dan berkolaborasi dalam pengembangan kode. Selain berfungsi sebagai repositori kode, GitHub juga menawarkan layanan GitHub *pages*, yang memungkinkan pengguna mempublikasikan website secara gratis dari repositori user [45]. GitHub memanfaatkan konsep *repositories* sebagai folder terpusat untuk menyimpan semua file, data, dan dokumentasi terkait proyek penelitian, sehingga

memungkinkan pengelolaan proyek secara terstruktur. Terdapat fitur seperti *issues* yang digunakan untuk merancang, mendokumentasikan, dan mendiskusikan eksperimen, mulai dari hipotesis hingga interpretasi data. Selain itu, GitHub mendukung penggunaan versi kontrol untuk memastikan bahwa setiap perubahan dalam data atau analisis terdokumentasi dengan baik, serta adanya integrasi dengan dengan software atau platform seperti *visual studio code* untuk pengeditan lokal [46]. GitHub platform juga menawarkan fitur tambahan seperti GitHub *actions* untuk otomatisasi alur kerja, keamanan berbasis *dependabot* untuk melacak kerentanan kode, dan kolaborasi tim sesuai kebutuhan organisasi.

#### **2.4.7 Draw.io**

Draw.io adalah perangkat lunak yang dapat diakses pengguna secara online dan gratis untuk membuat charts atau diagram yang mudah diakses melalui berbagai perangkat tanpa perlu menginstall. Arsitektur design antarmuka pengguna yang cukup simple sehingga mudah dipahami. Adapun komponen edit diagram disajikan secara independent dan fleksibel untuk diimplementasikan oleh pengguna dalam merancang diagram sesuai kebutuhan pengguna. Draw.io memiliki kemampuan pembuatan pemodelan, tata letak, serta memfasilitasi pembuatan diagram dengan edit style, pewarnaan, dan teknologi yang cukup lengkap. Fitur utamanya mencakup penyimpanan model yang dibuat dalam berbagai format, pembuatan berbagai diagram basis data seperti diagram UML, diagram ER, serta ketersediaan template untuk desain basis data, menampilkan hubungan antar tabel, dan fitur mengekspor diagram basis data dalam berbagai format seperti gambar, PDF, HTML, dan XML. Selain itu, platform online ini mendukung kolaborasi pengguna, memungkinkan kerjasama tim dalam pembuatan dan penyuntingan diagram [47].

#### **2.4.8 Canva**

Canva adalah perangkat lunak desain grafis yang intuitif dan fleksibel berbasis *web* maupun mobile app yang dirancang untuk memudahkan pengguna dalam pembuatan berbagai materi visual dari gambar hingga video baik untuk pemula maupun profesional. Canva menggunakan antarmuka *drag-and-drop* yang

intuitif, dilengkapi dengan ribuan template, font, ikon, gambar, maupun video yang dapat diakses secara gratis atau melalui langganan premium untuk akses fitur yang lebih lengkap. Fitur unggulannya meliputi kemampuan membuat desain untuk media sosial seperti Instagram stories, brosur, kartu nama, video presentasi, iklan, hingga grafik data. Canva juga mendukung kolaborasi tim dengan fitur berbagi proyek dan penyimpanan berbasis *cloud*. Selain itu, hasil desain dapat diunduh dalam berbagai format seperti JPEG, PNG, PDF, MP4 Video, dan lainnya untuk keperluan digital maupun cetak [48].

#### 2.4.9 Google Drive

Google drive merupakan salah satu layanan yang disediakan oleh Google sebagai platform layanan penyedia *cloud* yang fleksibel sebagai media penyimpanan dan manajemen data. Google memungkinkan pengguna menyimpan, mengakses, dan berbagi file dari berbagai perangkat. Google Drive menyediakan ruang penyimpanan gratis sebesar 15 GB mana pengguna dapat memperluas penyimpanan dengan berlangganan paket Google One. Disisi lain peran google drive sangat mendukung dalam pemodelan komputasi modern seperti *machine learning*. Google drive dapat diintegrasikan langsung dengan me-mounting pada lingkungan Google Colaboratory (Colab) yang dapat berfungsi sebagai hard drive eksternal untuk proses pemodelan sebagai penyimpan data. Hal ini memudahkan proses pemodelan untuk memuat *dataset* berukuran besar yang tersimpan di drive dan kemudian menyimpan model kembali ke drive. Maka dari itu, dengan pemanfaatan google drive ini akan membuat efisiensi memori dan penyimpanan model menjadi lebih mudah dan otomatis lebih efektif.

### **III. METODE PENELITIAN**

#### **3.1 Waktu dan Tempat Penelitian**

##### **3.1.1 Waktu Penelitian**

Penelitian ini dilaksanakan selama kurang lebih 5 bulan dari bulan Februari hingga bulan Juni tahun 2025. Penelitian ini mengimplementasikan model algoritma *Convolutional Neural Network (CNN)* untuk klasifikasi 4 jenis atau *spesies* citra lebah madu tanpa sengat (*stingless*) pada Lembah Suhita. Berikut adalah tabel rincian estimasi waktu pelaksanaan penelitian ini.

Tabel 3. 1. Waktu Pelaksanaan Penelitian

No	Kegiatan Penelitian	Februari				Maret				April				Mei				Juni			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	<i>Planning</i>																				
2	<i>Data Collect</i>																				
3	Eksplorasi Dataset																				
4	<i>Preprocessing Data</i>																				
5	<i>Data Augmentasi</i>																				
6	<i>Implementasi CNN</i>																				
7	<i>Training Model</i>																				
8	<i>Evaluasi Model</i>																				
9	<i>Penulisan Tugas Akhir</i>																				

##### **3.1.2 Tempat Penelitian**

Penelitian ini dilaksanakan di lingkungan Laboratorium Komputer, Program Studi Teknik Infomatika, Universitas Lampung. Adapun data penelitian berupa empat jenis (*spesies*) citra lebah madu tanpa sengat (*stingless*) diambil dari Lembah Suhita yang berlokasi di Jalan. Batin Mangku Negara, Kelurahan. Batu Putuk, Kecamatan. Teluk Betung Utara, Kota Bandar Lampung.

## 3.2 Alat dan Bahan Penelitian

### 3.2.1 Alat Penelitian

Pada penelitian ini, berikut beberapa alat yang digunakan untuk mendukung setiap tahapan yang tepat dan sesuai dengan prosedur penelitian. Adapun alat pada penelitian ini terbagi menjadi dua jenis, yaitu perangkat keras dan perangkat lunak yang diuraikan pada tabel berikut.

Tabel 3. 2. Perangkat Keras

Perangkat Keras	Fungsi
Laptop Intel Core i7 Gen 11th RAM 16GB SSD 512GB	Melakukan perancangan, pengolahan dataset, dan implementasi model klasifikasi citra
Mouse	Menggerakkan kursor
Keyboard	Mengetikkan input kata, kalimat, atau simbol
Smartphone Oppo Reno 6 5G	Mengambil data citra gambar lebah madu
Lensbong Macro 42mm	Mengoptimalkan pengambilan objek citra lebah
Flashdisk	Menyimpan data citra lebah madu ( <i>backup file</i> )

Tabel 3. 3. Perangkat Lunak

Perangkat Lunak	Keterangan
Google Chrome	<i>Platform</i> pendukung dalam <i>browsing</i> internet
Google Colab (spesifikasi <i>runtime</i> GPU T4)	<i>Platform</i> untuk pengolahan dataset dan implementasi <i>training</i> model
Jupyter Notebook	<i>Platform</i> untuk olah data tahap <i>pre-processing</i>
Google Drive	Media <i>backup coding</i> dari Google Colab
Microsoft Word	<i>Tool</i> untuk mengedit laporan dan dataset
Draw.io	<i>Tool</i> untuk membuat desain diagram
Canva	<i>Tool</i> multimedia untuk <i>editing</i> gambar/grafis
GitHub	Media untuk <i>upload</i> perubahan/tahapan <i>coding</i>
Python	Bahasa pemrograman implementasi model

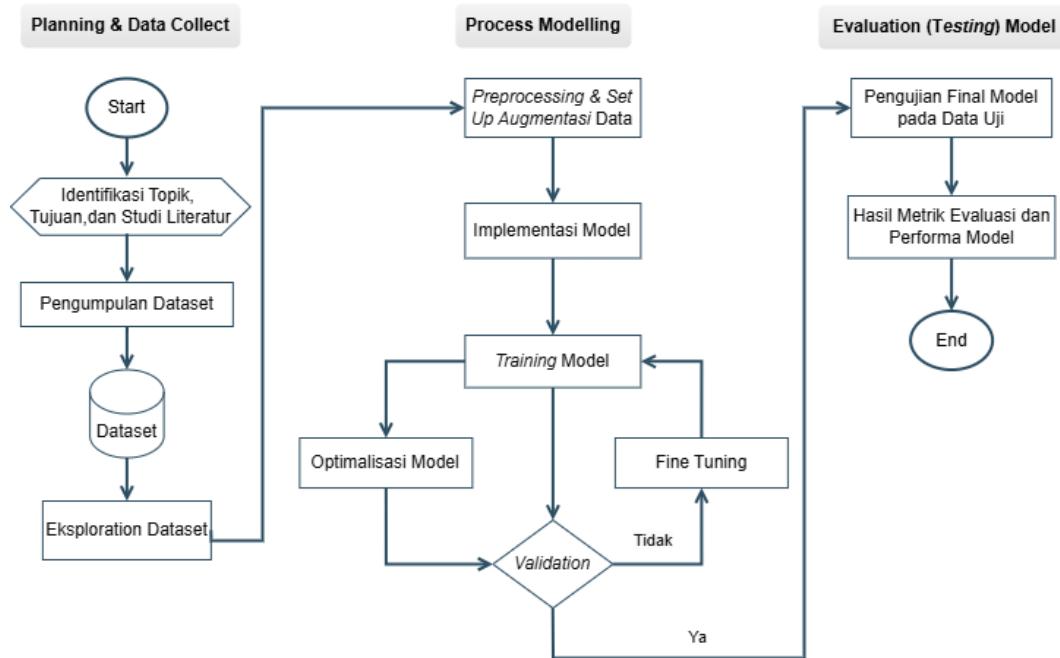
### 3.2.2 Bahan Penelitian

Pada penelitian ini terdapat beberapa bahan penelitian yang digunakan untuk mendukung informasi, metode, dan proses pengembangan model.

1. Data citra lebah madu tanpa sengat (*stingless*) pada Lembah Suhita.
2. Referensi jurnal penelitian yang digunakan sebagai acuan implementasi pemodelan *Convolutional Neural network (CNN)*.
3. Analisis penelitian terkait teknik augmentasi data dan evaluasi dalam pemodelan klasifikasi.

### 3.3 Tahapan Penelitian

Penelitian ini dilakukan dalam 3 tahapan utama mencakup tahap *planning* dan data *collect*, pemodelan, dan evaluasi model dengan *testing* data. Adapun rincian tahapan proses penelitian ini dijelaskan pada Gambar 3.1., berikut.



Gambar 3. 1. Alur Kerja Proses Penelitian

### 3.3.1 Planning (Perencanaan)

Pada tahap perencanaan dilakukan persiapan terkait pengumpulan data dan studi literatur mencakup identifikasi masalah, tujuan penelitian, serta langkah-langkah sistematis dalam pengembangan model klasifikasi citra spesies lebah madu tanpa sengat (*stingless*) dengan arsitektur *deep learning* menggunakan *Convolutional Neural Network* (CNN). Pada tahap ini, ditentukan metode yang akan digunakan, pemilihan dataset, serta perangkat lunak maupun perangkat keras yang akan digunakan dalam mendukung penelitian model klasifikasi ini. Tahap ini juga mencakup identifikasi proses pengumpulan data, teknik pengolahan data, eksplorasi dataset, *preprocessing*, hingga evaluasi model. Dalam tahap ini, ditentukan bahwa pelatihan model akan dilakukan dengan mempertimbangkan performa model pretrained CNN dengan menggunakan teknik augmentasi data untuk meningkatkan variasi data dan meningkatkan akurasi model berdasarkan temuan literatur sebelumnya.

### 3.3.2 Data Collect (Pengumpulan Data)

Tahapan selanjutnya yang dilakukan dalam penelitian ini yaitu pengumpulan dataset. Pengumpulan dataset ini dilakukan dengan mengambil citra lebah madu secara langsung di Lembah Suhita menggunakan kamera *smartphone* dengan tambahan lensa makro eksternal untuk mendukung pengambilan citra lebah menjadi lebih jelas. Proses ini dilakukan dengan mempertimbangkan kondisi pencahayaan dan sudut pengambilan agar citra yang dihasilkan memiliki kualitas cukup baik untuk diolah pada tahap selanjutnya. Pada proses pengambilan data, citra lebah madu yang diambil akan dikelompokkan berdasarkan jenis atau *spesies* sebanyak empat jenis lebah madu *stingless* dan dijadikan label dalam proses pelatihan model *Convolutional Neural Network* (CNN). Lebah madu yang diambil untuk pelatihan model dalam penelitian ini terdiri dari genus *tetrigona sp.*, *heterotrigona sp.*, *tetragonula sp.*, dan *geniotrigona sp.*. Keempat jenis lebah ini biasa dikenal dengan nama lebah kelulut, klanceng, dan bala-bala. Adapun label asli setiap kelas citra lebah yang diambil divalidasi oleh pemilik Suhita Bee Farm sebagai lokasi pengambilan data citra yaitu Ibu Isnina yang sudah merintis peternakan lebah dengan nama PT. Suhita Lebah Indonesia sejak tahun 2016, selain

itu validasi citra juga didasarkan pada *spot* pengambilan data citra dari masing-masing pintu masuk sarang lebah yang memiliki karakteristik berbeda-beda setiap jenisnya, serta label identitas jenis lebah yang dapat dilihat pada lampiran 6. Jumlah dataset citra lebah madu tanpa sengat (*stingless*) pada setiap kelas berjumlah seimbang agar tidak mempengaruhi performa model saat pelatihan. Jumlah pengumpulan data ini dilakukan sebanyak 200 data citra untuk satu jenis lebah, sehingga diperoleh sekitar 800 citra lebah madu *stingless* sesuai kategori kelas label. Kemudian, selanjutnya data yang telah dikumpulkan akan dieksplorasi dari segi kualitas, proporsi data, validasi data, *renaming*, *foldering*, *resizing*, dan *set-up* awal untuk pelatihan model.

### 3.3.3 Eksplorasi dan *Pre-processing* Data

Tahap berikutnya setelah dataset terkumpul yaitu eksplorasi dataset yang bertujuan untuk memahami karakteristik data sebelum masuk ke proses berikutnya. Pada tahap ini, peneliti melakukan beberapa proses mencakup pemeriksaan jumlah gambar dalam setiap kelas data, analisis kualitas gambar atau potensi noise, nama setiap data citra, atribut data citra, maupun data yang tidak relevan dari tiap kelas label data yang sudah diambil. Untuk melakukan *checking* dataset, citra dari setiap kelas *spesies* lebah madu *stingless* akan dianalisis secara keseluruhan data untuk melihat pola unik yang dapat berpengaruh terhadap prediksi model nantinya. Selain itu, akan dilakukan analisis untuk mengetahui distribusi atau proporsi keseluruhan kelas dataset, sehingga jika ditemukan ketidakseimbangan antar kelas, maka akan dilakukan penyesuaian untuk mengembangkan dan memperbaiki proporsi data tiap kelas. Lalu, tahapan pada *preprocessing* data juga bertujuan untuk menyiapkan data agar lebih optimal dalam proses pelatihan model *Convolutional Neural Network* (CNN). Pada tahap ini, peneliti akan melakukan *resizing* gambar ke ukuran yang sama yaitu 224x224 piksel, agar sesuai dengan *default* model *pre-trained* yang akan diterapkan. Lalu, dilakukan normalisasi nilai piksel untuk mempercepat komputasi pemodelan selama *training*. Data akan melalui tahap label data menjadi numerikal melalui *label hot-encoding*. Data citra yang telah di *pre-processing* akan disimpan dalam format .png dalam folder terpisah sesuai kelas label data sebelumnya. Terakhir, dataset dibagi sesuai folder masing-masing menjadi data latih

(*training*), validasi (*validation*), dan uji (*testing*) dengan perbandingan 80% : 10% : 10%.

### 3.3.4 Augmentasi Data

Selanjutnya dalam pelatihan model akan dilakukan teknik *augmentasi* yang akan dilakukan secara *on the fly* selama pelatihan (*training*) berlangsung. Adapun teknik augmentasi yang digunakan meliputi *rotation*, *zoom*, *shear range*, *flip horizontal* dan *vertikal*. Teknik *augmentasi* yang diterapkan meliputi *flipping* untuk membalik gambar, *rotation* untuk memutar gambar dalam sudut tertentu, serta *zooming* untuk memperbesar bagian tertentu dari gambar. Teknik lanjutan lainnya seperti *brightness adjustment* dan *cropping* juga diterapkan untuk meningkatkan *robust* model tertentu jika dibutuhkan. Teknik augmentasi yang diterapkan selama pelatihan dengan *Image Data Generator* pada *TensorFlow* atau *Albumentations* merupakan teknik tambahan yang sudah dalam penelitian sebelumnya dapat meningkatkan variasi data dan mempengaruhi akurasi akhir atau hasil model. Dengan teknik *augmentasi*, model *Convolutional Neural Network* (*CNN*) diharapkan dapat mempelajari pola yang lebih beragam sehingga mampu meningkatkan *generalisasi* sehingga model tidak *underfitting* dan mengurangi kemungkinan terjadinya *overfitting* dalam pelatihan model.

### 3.3.5 Implementasi *Pretrained CNN Model*

Setelah dataset siap dan *set-up* awal program dan teknik augmentasi dilakukan, maka selanjutnya akan dilakukan implementasi *Convolutional Neural Network* (*CNN*) dimulai dengan menerapkan arsitektur *pretrained* model *CNN* yaitu EfficientNetB0 dan MobileNetV2. Model *pre-trained* yang digunakan dalam penelitian ini ditentukan berdasarkan karakteristik model yang dikenal sebagai model dengan komputasi yang ringan dan efektif pada dataset yang jumlahnya terbatas. Selain itu bobot dari parameter kedua model ini juga lebih ringan untuk integrasi lanjutan pada tahap *deployment* model. Penerapan arsitektur model ini dilakukan berdasarkan teori *transfer learning* dari Tensorflow Keras. Kedua model akan digunakan sebagai *feature extractor* pada model awal tanpa pelatihan dengan meng-*freeze* semua lapisan konvolusional untuk mengevaluasi kemampuan dasar

setiap model dalam mengklasifikasikan data citra lebah madu *stingless* yang akan dilatih. Lalu pada bagian *top layer* klasifikasinya akan ditambahkan *custom head* berupa *Global Average Pooling 2D* untuk reduksi dimensi, dan *fully connected layer* berupa *Dense layer* dan regularisasi *Dropout* (0.5), diikuti *output layer* berjumlah 4 neuron dengan fungsi aktivasi *softmax*. Lapisan ini disesuaikan dengan jumlah kelas data spesies lebah, terdiri dari *Global Average Pooling*, *Dense layer*, *Dropout*, dan *Output layer*. Hasil akurasi selama training akan dipantai dengan nilai *Categorical Cross-Entropy* dengan *optimizer* Adam dengan *learning rate* awal 0.001. Jika akurasi belum optimal, akan dilakukan proses *fine tuning* dengan membuka beberapa lapisan akhir model *pre-trained* dan melatih ulang lapisan terakhir lebih dalam dengan *learning rate* lebih rendah. Implementasi dilakukan di TensorFlow Keras pada Google Colab dengan spesifikasi *runtime* GPU T4 yang tersedia dan bahasa pemrograman Python untuk pengolahan data hingga implementasi pelatihan model. Update tahapan proses pelatihan model akan di *upload* pada *Github* dan *Google Drive*. Selanjutnya model akan dilatih pada proses training untuk validasi dan evaluasi.

### 3.3.6 Pelatihan (*Training*) Model

Pelatihan model *pretrained* dari kedua model yaitu *MobileNetV2* dan *EfficientNetB0* akan dilakukan dalam dua fase. Fase pertama melatih lapisan *custom head* klasifikasi dengan layer *Global Average Pooling*, *Dense* dan *Dropout* dengan lapisan konvolusional tetap dibekukan, menggunakan *optimizer* Adam dengan *batch size* 10 - 64, dan penerapan teknik augmentasi data meliputi *rotation*, *flipping*, *zoom*, dan *shear*. Pelatihan akan diuji selama 30-50 epoch. Selama pelatihan akan diterapkan metrik akurasi meliputi *accuracy*, *loss*, *precision*, *recall* yang dipantau secara berkala pada data validasi. Dengan adanya proses ini bertujuan untuk memungkinkan deteksi awal *overfitting* dan penyesuaian *hyperparameter* seperti *learning rate* dan parameter terkait. Selain itu dapat digunakan fungsi *callback* seperti *Early Stopping* dan *Model Checkpoint* untuk menyimpan model terbaik dan menghentikan pelatihan jika akurasi validasi mulai menurun atau *stagnan* untuk mencegah model terlalu menghafal data pelatihan. Pada penelitian ini akurasi diperkirakan pada angka 70-80%, karena persentase

akurasi ini artinya model sudah dikatakan cukup baik, namun untuk hasil yang lebih akurat maka dapat dilakukan proses *fine tuning* pada 100-200 lapisan akhir model *pretrained* pada blok konvolusi akhir yang akan dibuka (*unfreeze*), dan pelatihan dilanjutkan dengan *optimizer* Adam dengan *learning rate* yang diturunkan. Pelatihan dilanjutkan untuk melihat performa akurasi pelatihan dan validasi, hingga diperoleh hasil akurasi model tertinggi.

### 3.3.7 Evaluasi Model

Setelah tahapan *training* selesai, tahap terakhir yang dilakukan yaitu evaluasi model tertinggi yang didapatkan saat proses *training* dengan menguji model pada data uji yang sudah dibagi pada persiapan awal dataset, yang mana data ini belum pernah digunakan selama pelatihan dan validasi model. Tujuan evaluasi model ini yaitu digunakan untuk mengukur kemampuan akurasi model pada data baru dan kemampuan generalisasi model. Hasil prediksi yang dihasilkan akan ditampilkan dengan mengukur metrik akurasi seperti *accuracy* untuk keseluruhan persentase prediksi benar dan *loss* untuk persentase kesalahan prediksi model digunakan sebagai indikator performa keseluruhan model terhadap data uji. Selain itu untuk analisis setiap kelas, dilakukan perhitungan value dari nilai *precision*, *recall*, dan *F1-score* dengan pendekatan *macro average* untuk memastikan semua kelas *spesies* lebah dievaluasi secara seimbang yang akan ditampilkan melalui *classification report* hasil evaluasi model. Lalu, divisualisasikan hasil perolehan *confusion matrix* untuk mengidentifikasi pasangan kelas yang sering tertukar serta pola kesalahan seperti *false positive(FP)* dan *false negative(FN)* dominan dari kelas data menggunakan *library scikit learn*.

## **V. PENUTUP**

### **5.1 Kesimpulan**

Berdasarkan hasil dan pembahasan pada penelitian yang telah dilakukan, maka disimpulkan bahwa:

1. Dari kedua model *deep learning* CNN MobileNetV2 dan EfficientNetB0 yang dilatih, model terbaik yang mampu mengklasifikasi spesies *stingless bee* yaitu arsitektur model MobileNetV2 dengan nilai akurasi tertinggi yang dicapai sebesar 91.25%.
2. Penambahan variabel teknik augmentasi data pada kedua arsitektur model klasifikasi spesies *stingless bee* ini mampu meningkatkan akurasi model sebesar 1% pada model MobileNetV2 dan 24% pada model EfficientNetB0.
3. Selama proses *fine tuning*, hyperparameter dengan *batch size* 16, *epoch* 50 hingga 100, dan *learning rate* 0.00001 merupakan hyperparameter terbaik yang memperoleh akurasi tertinggi pada model MobileNetV2.

### **5.2 Saran**

Berdasarkan hasil dan temuan penelitian yang telah dilakukan, berikut saran untuk penelitian sejenis:

1. Meningkatkan jumlah kelas spesies dan dataset citra *stingless bee* agar model dapat digunakan secara general pada klasifikasi spesies *stingless bee* yang lebih kompleks.
2. Meningkatkan kualitas citra pada dataset dengan pengambilan gambar yang lebih *profesional* untuk objek dinamis seperti lebah agar model dapat menangkap detail citra dengan lebih akurat.
3. Menambahkan kelas ‘anomali’ pada *training* model untuk meningkatkan kemampuan model secara general dan dapat dilanjutkan integrasi tahap *deployment* model pada *mobile app* untuk praktis penggunaan.

## DAFTAR PUSTAKA

- [1] L. V Dicks *et al.*, “A global assessment of drivers and risks associated with pollinator decline,” *Nat Ecol Evol*, vol. 5, no. 2, pp. 145–154, 2021, doi: 10.1038/rs.3.rs-90439/v1.
- [2] S. Avila and others, “Stingless bee honey: Quality parameters, bioactive compounds, health-promotion properties and modification detection strategies,” *Trends Food Sci Technol*, vol. 81, pp. 37–50, 2018, doi: 10.1016/j.tifs.2018.09.002.
- [3] R. Kumar, A. Thakur, S. Kumar, and Y. A. Hajam, “Royal jelly a promising therapeutic intervention and functional food supplement: A systematic review,” *Heliyon*, 2024, doi: 10.1016/j.heliyon.2024.e37138.
- [4] D. Katiyar, “Propolis: A natural biomaterial,” *Mater Today Proc*, 2023, doi: 10.1016/j.matpr.2023.05.522.
- [5] M. Tripathi, “Analysis of convolutional neural network based image classification techniques,” *Journal of Innovative Image Processing (JIIP)*, vol. 3, no. 2, pp. 100–117, 2021, doi: 10.36548/jiip.2021.2.003.
- [6] S. G. Paul *et al.*, “A real-time application-based convolutional neural network approach for tomato leaf disease classification,” *Array*, vol. 19, p. 100313, 2023, doi: 10.1016/j.array.2023.100313.
- [7] J. Khabibi, A. Albayudi, and D. J. Ginting, “Kualitas madu dari 3 spesies lebah penghasil madu: Honey quality from 3 species of honey producing bees,” *Jurnal Silva Tropika*, vol. 6, no. 1, pp. 43–50, 2022, doi: 10.22437/jsilvtrop.v6i1.21308.
- [8] Food and A. O. (FAO), “Pollinators in agriculture: A global assessment,” 2019.
- [9] C. Gruter, *Stingless bees: Their behaviour, ecology and evolution*. Springer, 2020.
- [10] E. S. Eriana and A. Zein, “Artificial intelligence (AI),” 2023.
- [11] S. Rifky and others, *Artificial intelligence: Teori dan penerapan AI di berbagai bidang*. PT. Sonpedia Publishing Indonesia, 2024.
- [12] I. Sulistyowati, *Buku ajar mata kuliah kecerdasan buatan (artificial intelligence)*. Umsida Press, 2021.

- [13] W. Setiawan, *Deep learning menggunakan convolutional neural network: Teori dan aplikasi*. Media Nusa Creative (MNC Publishing), 2020.
- [14] J. D. Kelleher, *Deep learning*. The MIT Press, 2019.
- [15] F. Chollet, *Deep learning with Python*, 2nd ed. Manning Publications Co., 2021.
- [16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.
- [17] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, and L. Farhan, “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,” *J Big Data*, vol. 8, pp. 1–74, 2021, doi: 10.1186/s40537-021-00444-8.
- [18] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into deep learning*. Cambridge University Press, 2023.
- [19] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artif Intell Rev*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.
- [20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.
- [21] M. Tan and Q. V Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning*, in Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 6105–6114. [Online]. Available: <http://proceedings.mlr.press/v97/tan19a.html>
- [22] A. Dzulkarnain and M. N. P. Ma’ady, “Teori dan penerapan backpropagation neural networks untuk internet of things: Online dan batch mode,” *SISFO*, vol. 11, no. 1, pp. 25–38, 2024.
- [23] I. D. Id, *Machine learning: Teori, studi kasus dan implementasi menggunakan Python*, vol. 1. Unri Press, 2021.
- [24] R. Y. Sun, “Optimization for deep learning: An overview,” *Journal of the Operations Research Society of China*, vol. 8, no. 2, pp. 249–294, 2020, doi: 10.1007/s40305-020-00309-6.
- [25] H. Fitriyah and R. C. Wihandika, *Dasar-dasar pengolahan citra digital*. Universitas Brawijaya Press, 2021.

- [26] I. Wulandari, H. Yasin, and T. Widiharih, “Klasifikasi citra digital bumbu dan rempah dengan algoritma convolutional neural network (CNN),” *Jurnal Gaussian*, vol. 9, no. 3, pp. 273–282, 2020, doi: 10.14710/j.gauss.9.3.273-282.
- [27] F. Marpaung, F. Aulia, and R. C. Nabila, “Computer vision dan pengolahan citra digital,” 2022.
- [28] J. Howard and S. Gugger, *Deep learning for coders with Fastai and PyTorch*. O’Reilly Media, 2020.
- [29] I. Rianto and I. P. I. Santosa, *Data preparation untuk machine learning & deep learning*. Penerbit Andi, 2025.
- [30] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, “Image data augmentation for deep learning: A survey,” 2022, *ArXiv*. doi: 10.48550/arXiv.2204.08610.
- [31] M. F. Gunardi, “Implementasi augmentasi citra pada suatu dataset,” *Jurnal Informa*, vol. 9, no. 1, pp. 1–5, 2023.
- [32] J. M. Valiente, M. Juan-Borrás, F. López-García, and I. Escriche, “Automatic pollen recognition using convolutional neural networks: The case of the main pollens present in Spanish citrus and rosemary honey,” *Journal of Food Composition and Analysis*, vol. 123, p. 105605, 2023, doi: 10.1016/j.jfca.2023.105605.
- [33] W. Kelley, I. Valova, D. Bell, O. Ameh, and J. Bader, “Honey sources: Neural network approach to bee species classification,” *Procedia Comput Sci*, vol. 192, pp. 650–657, 2021, doi: 10.1016/j.procs.2021.08.067.
- [34] K. Buschbacher, D. Ahrens, M. Espeland, and V. Steinhage, “Image-based species identification of wild bees using convolutional neural networks,” *Ecol Inform*, vol. 55, p. 101017, 2020, [Online]. Available: <https://www.researchgate.net/publication/33699958>
- [35] D. A. D. Junior and others, “Automatic method for classifying COVID-19 patients based on chest X-ray images, using deep features and PSO-optimized XGBoost,” *Expert Syst Appl*, vol. 183, p. 115452, 2021, doi: 10.1016/j.eswa.2021.115452.
- [36] D. Irfansyah and others, “Arsitektur convolutional neural network (CNN) Alexnet untuk klasifikasi hama pada citra daun tanaman kopi,” *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 6, no. 2, pp. 87–92, 2021, doi: 10.30591/jpit.v6i2.2802.

- [37] W. Bismi and M. Qomaruddin, “Klasifikasi citra genus *Panthera* menggunakan pendekatan deep learning berbasis convolutional neural network (CNN),” *Jurnal Informatika dan Rekayasa Perangkat Lunak*, pp. 172–179, 2023, doi: 10.36499/jinrpl.v5i2.8931.
- [38] A. Achmadi and others, “Identifikasi penyakit daun tembakau berbasis pengolahan citra dengan metode convolutional neural network (CNN) dan metode transfer learning,” *Jurnal Sistem Komputer dan Kecerdasan Buatan*, vol. 8, no. 1, pp. 10–16, 2024, doi: 10.47970/siskom-kb.v8i1.654.
- [39] A. Akram, K. Fayakun, and H. Ramza, “Klasifikasi hama serangga pada pertanian menggunakan metode convolutional neural network,” *Building of Informatics, Technology and Science (BITS)*, vol. 27, no. 09, 2023, doi: 10.47065/bits.v5i2.4063.
- [40] S. F. Chairunissa, “Klasifikasi Jenis Lebah Madu Menggunakan Metode Convolutional Neural Network,” *JITTER - Jurnal Ilmiah Teknologi dan Komputer*, vol. 4, no. 3, 2023, doi: 10.24843/JTRTI.2023.v04.i03.p04.
- [41] P. G. Naik, “Conceptualizing python in google colab: Hands-on practical sessions,” 2023, *Shashwat Publication*. [Online]. Available: <https://www.researchgate.net/publication/357929808>
- [42] A. Suharto, *Fundamental bahasa pemrograman Python*. Eureka Media Aksara, 2023.
- [43] A. J. Dhruv, R. Patel, and N. Doshi, “Python: The most advanced programming language for computer science applications,” Science and Technology Publications, Lda, 2021, pp. 292–299. [Online]. Available: <https://api.semanticscholar.org/CorpusID:234894687>
- [44] A. Geron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., 2022.
- [45] A. Arisantoso, F. R. Lithardo, M. Faisal, A. Sugianto, M. Z. Hakim, and S. Samadara, “Buku ajar pemrograman web dasar,” 2024.
- [46] K. Y. Chen, M. Toro-Moreno, and A. R. Subramaniam, “GitHub is an effective platform for collaborative and reproducible laboratory research,” 2024. doi: 10.48550/arXiv.2408.09344.
- [47] T. A. Vakaliuk, O. V Korotun, and S. O. Semerikov, “The selection of cloud services for ER-diagrams construction in IT specialists databases teaching,” in *CTE Workshop Proceedings*, Mar. 2021, pp. 384–397. doi: 10.55056/cte.295.

- [48] A. P. Gehred, “Canva,” *J Med Libr Assoc*, vol. 108, no. 2, p. 338, 2020, doi: 10.5195/jmla.2020.940.
- [49] J. Hang, D. Zhang, P. Chen, J. Zhang, and B. Wang, “Classification of plant leaf diseases based on improved convolutional neural network,” *Sensors (Switzerland)*, vol. 19, no. 19, p. 4161, 2019, doi: 10.3390/s19194161.
- [50] G. Kang, K. Liu, B. Hou, and N. Zhang, “3D multi-view convolutional neural networks for lung nodule classification,” *PLoS One*, vol. 12, no. 11, p. e0188290, 2017, doi: 10.1371/journal.pone.0188290.